PROBABILISTIC GRADIENTS FOR FAST CALIBRATION OF DIFFERENTIAL EQUATION MODELS *

JON COCKAYNE† AND ANDREW B. DUNCAN‡

Abstract. Calibration of large-scale differential equation models to observational or experimental data is a widespread challenge throughout applied sciences and engineering. A crucial bottleneck in state-of-the art calibration methods is the calculation of local sensitivities, i.e. derivatives of the loss function with respect to the estimated parameters, which often necessitates several numerical solves of the underlying system of partial or ordinary differential equations. In this paper we present a new probabilistic approach to computing local sensitivities. The proposed method has several advantages over classical methods. Firstly, it operates within a constrained computational budget and provides a probabilistic quantification of uncertainty incurred in the sensitivities from this constraint. Secondly, information from previous sensitivity estimates can be recycled in subsequent computations, reducing the overall computational effort for iterative gradient-based calibration methods. The methodology presented is applied to two challenging test problems and compared against classical methods.

Key words. PDE constrained optimisation, sensitivity analysis, probabilistic numerics

AMS subject classifications. 62G86, 62P30, 35Q93, 35R30

1. Introduction. Complex systems arising in applied sciences and engineering are often modelled by systems of coupled ordinary or partial differential equations (ODEs or PDEs) derived from the underlying physical principles. Typically, the specific model behaviour will depend on a vector of parameters which must be *calibrated* to observations of system. A major challenge in calibration is the high computational cost associated with numerically solving the mathematical model for a given value of the parameters. This is particularly relevant for large-scale models incorporating multi-physics and multiscale behaviour, as arise in the context of digital twins [49]. This high cost often precludes the use of many iterative methods for calibration, including both optimisation methods and Bayesian approaches that use sampling algorithms such as Markov chain Monte-Carlo (MCMC). Each of these requires at least one solve of the governing equations per iteration of the algorithm. In practice MCMC often requires of $\mathcal{O}(10^5)$ model evaluations [20].

The calibration of differential equation models to observed data can be formulated as a constrained optimisation problem [7, 22, 28], which is solved using deterministic or stochastic optimisation methods. Most fundamental optimisation methods[1] either require or are accelerated by access to derivatives of the functional to be minimised, so that the solver for the underlying equations must be augmented with a routine that provides the derivative of the solution with respect to model parameters. Employing an approximation of the gradient, such as a finite-difference approximation, may seem attractive due to ease of implementation, but obtaining accurate approximations can be challenging and, when the parameter dimension is large, computationally expen-

*Submitted to the editors September 3, 2020.

Funding: JC was supported by Wave 1 of The UKRI Strategic Priorities Fund under the EPSRC Grant EP/T001569/1, particularly the "Digital Twins for Complex Engineering Systems" theme within that grant, and The Alan Turing Institute. ABD was supported by the Lloyds Register Foundation Programme on Data Centric Engineering and by The Alan Turing Institute under the EPSRC grant [EP/N510129/1].

†The Alan Turing Institute (jcockayne@turing.ac.uk)

‡Imperial College London and The Alan Turing Institute (a.duncan@imperial.ac.uk)

[1] i.e. any method in the Newton family of optimisation methods.

sive. Thus it is usually preferable to obtain derivatives using *first-order sensitivity analysis*, which expresses the derivatives as the solution of an auxiliary system of differential equations known as the *sensitivity equations*. While the sensitivity equations are linear, they depend on the solution of the underlying equations and so must typically be solved numerically. Thus, computing the sensitivities is at least as expensive as solving the system itself. Further, sensitivities must be computed for every parameter value at which a gradient evaluation is required, making them prohibitively expensive for use in optimisation methods, where gradients are typically required over a large sequence of parameter values.

In the context of model calibration and uncertainty quantification, Gaussian processes (GPs) are often used as surrogate models for the solution of the underlying equations with the aim of making the calibration of such models tractable [27, 52]. This approach is advantageous as derivatives of the GP posterior mean can usually be computed explicitly, permitting the use of gradient based optimisation methods and sampling. While GP surrogate models do provide an effective approach to calibrating black-box computer codes where little is known about the structure of the underlying model, this comes at the price of *data-efficiency*. Information about the gradient can only be obtained from multiple function evaluations near to the location of the required gradient, so again numerous evaluations may be required, particularly if the dimension of the parameter space is high.

These highlighted issues motivate the novel approach to computing sensitivities for optimisation problems presented in this paper. Our proposed approach is able to bridge the gap between the classical approach of numerically solving the sensitivity equations and the purely data-driven surrogate model approach. This is achieved by introducing a nonparametric Gaussian process model for the solution of the sensitivity equations that is defined over the entire parameter space. The output is a posterior distribution on the space of vector fields in parameter space, whose mean can be interpreted as an estimate of the local sensitivity across multiple parameter locations and whose variance controls the error in this estimate under regularity assumptions.

This approach offers various advantages to the state-of-the-art approaches: firstly the computational cost of the method can be carefully controlled by the user, either to attain a desired level of accuracy as measured by the "width" of the posterior distribution or to fit within a given fixed computational budget. Secondly, estimates of gradients at multiple parameter locations are able to share information between them to provide accurate gradient approximations without necessitating additional numerical solves of the underlying PDE model. Thirdly, the posterior distribution can be efficiently updated when a gradient evaluation at a new parameter value is required. These three advantages are particularly pertinent to model calibration methods which require multiple gradient evaluations along a trajectory.

Besides the immediate application to calibration of PDE models, the efficient approximation of sensitivities for large scale PDE models is of independent interest, with wide ranging applications including model order reduction [42], shape optimisation [33] and uncertainty quantification [2]. The probability distribution output from our new approach has a rigorous Bayesian interpretation, allowing it to be composed within inference and computation pipelines in a coherent manner to enable propagation of uncertainty.

1.1. Related Work. ODE- or PDE-constrained optimisation problems are a class of control problem in which the cost function involves the solution of a partial differential equation posed on a domain $D \subseteq \mathbb{R}^d$. Classically, such optimisation

problems arise in the context of design and control of engineering systems, for example in optimal topological design, shape design and optimal control of dynamic systems. See [26] for a review of optimisation algorithms for use in this context. Further, these problems arise naturally in the context of Bayesian inverse problems and model calibration. In particular, variational approaches to data assimilation for weather prediction can be naturally rephrased as PDE-constrained optimisation problems [18].

Sensitivity analysis seeks to quantify the dependence of a function $g(p)$ on perturbations of the problem data or parameters $p \in P$. Broadly speaking, we distinguish between *global* sensitivity analysis, which quantifies how input variability influences output variability of a model, and *local* sensitivity analysis which assesses the influence of infinitesimal input perturbations on model output. The former is typically assessed in terms of variance, classically using variants of Sobol' indices [51]. By contrast, local sensitivity analysis involves the calculation of partial derivatives of function outputs with respect to parameters. Local sensitivity analysis plays a fundamental role in the context of ODE- or PDE-constrained optimisation [9]. In this setting, let $g(u, p)$ denote the real-valued objective function for the optimisation problem, that depends on the solution $u(p)$ of a differential equation for a given parameter value $p \in P$. Then we seek to compute the total derivative $\frac{\mathrm{d}g}{\mathrm{d}p}(p)$, which constitutes the local sensitivities.

Generally speaking there are two approaches to computing such derivatives: the *forward* or *direct* method and the *adjoint* method. In the forward method, supposing that $p \subseteq \mathbb{R}^m$, the underlying equations are differentiated with respect to $p_1, \ldots, p_m$ to obtain a system of m equations for the sensitivities. The adjoint method originates in the theory of Lagrange multipliers in optimisation, and involves solving an auxiliary adjoint equation for the Lagrange multiplier λ from which the sensitivities can be directly computed. Given that the forward approaches involves solving a system of m equations while the adjoint approach involves solving only a single equation, the latter approach can be significantly more efficient for large m [48].

The computational cost of solving optimisation problems involving large-scale ODE or PDE models has motivated the use of *surrogate models*; approximations of the underlying model that have significantly lower computational overhead. Proposed approaches include using reduced order modelling based on reduced basis methods or proper orthogonal decompositions [4, 5]. These surrogate approaches are motivated by the fact that the adjoints, and therefore the gradients, of the low-dimensional surrogate model can obtained efficiently. Recent efforts involve combining neural network models with low-dimensional physical models to obtain efficient and accurate surrogate models [16, 46, 47, 24, 50]. Again, these methods exploit the fact that gradients of neural network models can be obtained efficiently through back-propagation.

Gaussian processes (GPs) have been widely used to provide black-box emulation of computationally expensive codes [45], with [30] providing a mature Bayesian formulation to the methodology. Emulation methods based on GPs are now widespread and find uses in numerous applications ranging from computer code calibration [27], uncertainty analysis [35] and MCMC [31, 12]. Among the first papers to consider application of emulation within sensitivity analysis was [36], which extended the work of [30] to computation of variance-based global sensitivities. Subsequent work by [29] considered a similar approach that exploited a tensor-product kernel to simplify the integration problems required, though this work did not consider the posterior covariance in their estimator. See [10] for a more extensive review of emulation-based global sensitivity analysis techniques, and [21, 3, 44] for a survey of applications of such approaches. One could envisage an analogous emulation strategy for local sensitivity analysis of computationally expensive models that involves first constructing

an emulator $\hat{g}$ of the objective function g and then evaluating the derivative $\frac{\mathrm{d}\hat{g}}{\mathrm{d}p}(p^*)$ which, assuming a conducive emulator, can be computed at a lower cost than the derivative of g itself. A notable disadvantage of this approach is that to approximate local sensitivities in this way would require *global* information about g, since unless a highly structured surrogate model is used little information can be obtained about $\frac{\mathrm{d}g}{\mathrm{d}p}(p)$ from the single evaluation $g(p)$.

The method proposed in this paper aims to bridge the gap between classical numerical approaches and emulation-based approaches to calculating sensitivities within optimisation problems. The proposed method can be interpreted as a Bayesian probabilistic numerical method [14] for the solution of the forward or adjoint sensitivity equations over $D \times P$. It is similar to the probabilistic meshless methods for solutions of PDEs presented in [13, Chapter 5], but extended across parameter space. This formalism presents several advantages. Firstly it permits a high level of adaptivity, in that the solution can be refined over both P and D to increase accuracy either globally over parameter space P, or locally for particular value of the parameters $p \in P$. Secondly, subject to regularity assumptions, estimates of the gradient at a parameter p may re-use information from nearby gradient evaluations, exploiting the smoothness of the sensitivity equations to reduce the computational effort required for accurate gradient estimates at p when nearby gradients have already been evaluated. Thirdly, gradient estimates can be updated efficiently, allowing the adaptivity and smoothness properties mentioned to be exploited within algorithms that depend upon local sensitivities, such as gradient-based optimisation algorithms.

1.2. Contributions. The main contributions of the paper are as follows:

- We develop a probabilistic framework for computing gradients for differential equation models.
- We study the theoretical properties of this method, in particular its robustness to discretisation error.
- We demonstrate how the inferred gradients can be leveraged in optimisation problems.
- The results are demonstrated on a number of model problems to analyse its performance in comparison to classical approaches.

1.3. Structure of the Paper. The paper proceeds as follows. In section 2 the classical approach to computing local sensitivities is formulated with examples of application to the problem of computing sensitivities for a simple PDE. section 3 presents the novel probabilistic approaches and provides theoretical results relating to their accuracy and stability. section 4 discusses the use of the probabilistic methods introduced in optimisation problems, and the empirical performance of these methods is assessed in section 5. We conclude with some discussion in section 6. The supplementary material contains the proofs required for the paper in section S1.

1.4. Notation. Let $W^{k,p}(D)$ denote the Sobolev space in which each function has k weak derivatives with finite $L^p(D)$ norm. We will use the notation $H^k(D) = W^{k,2}(D)$. Further let $H^k_0(D)$ denote the subset of $H^k(D)$ for which all $f \in H^k_0(D)$ have $f = 0$ on ∂D and $H^{-k}(D)$ to be the dual of $H^k_0(D)$. For two normed spaces $\mathcal{U}$, $\mathcal{V}$ we will use the notation $\mathcal{L}(\mathcal{U}, \mathcal{V})$ to denote the set of all bounded linear operators from $\mathcal{U}$ to $\mathcal{V}$. For the set of all bounded linear functionals on $\mathcal{U}$ we will use the notation $\mathcal{U}^* = \mathcal{L}(\mathcal{U}, \mathbb{R})$. When $\mathcal{U}$ is a set of functions on some domain D we will use the notation $\delta[x]$ to denote the evaluation functional for the point $x \in D$, i.e. $\delta[x](u) = u(x)$.

When both $\mathcal{U}$ and $\mathcal{V}$ are Hilbert spaces, for an operator $A \in \mathcal{L}(\mathcal{U},\mathcal{V})$ let $A^\dagger \in \mathcal{L}(\mathcal{V},\mathcal{U})$ denote the adjoint of A. For $A \in \mathcal{L}(\mathcal{U},\mathcal{U})$, recall that the trace of A is defined as $\text{trace}(A) = \sum_{i=1}^{\infty} \langle Ae_i, e_i\rangle$ where $(e_i)_{i=1}^{\infty}$ is an arbitrary orthonormal basis of $\mathcal{U}$.

Several operator norms will be required. For an operator $A : \mathcal{U} \to \mathcal{V}$ we will denote the operator norm by $\|A\|_{\mathcal{U}\to\mathcal{V}} = \sup_{u\in\mathcal{U}} \|Au\|_{\mathcal{V}}/\|u\|_{\mathcal{U}}$. When $\mathcal{U} = \mathcal{V}$ we will simply use the notation $\|A\|_{\mathcal{U}}$. The trace norm is given by $\|A\|_{\mathsf{tr}} = \text{trace}([A^\dagger A]^{\frac{1}{2}})$ while the Hilbert-Schmidt norm is given by $\|A\|_{\mathsf{HS}} = \text{trace}(A^\dagger A)^{\frac{1}{2}}$. Recall that $\|A\|_{\mathcal{U}\to\mathcal{V}} \leq \|A\|_{\mathsf{HS}} \leq \|A\|_{\mathsf{tr}}$.

1.4.1. Fréchet Derivatives. Of central importance to the paper is the concept of a Fréchet derivative. Let $\mathcal{U}$ and $\mathcal{V}$ each be normed spaces and consider a function $f : \mathcal{U} \to \mathcal{V}$. When it exists, Fréchet derivative of f at $u \in \mathcal{U}$ is defined to be the operator $\frac{\mathrm{d}f}{\mathrm{d}u}[u] \in \mathcal{L}(\mathcal{U},\mathcal{V})$ that satisfies

$$\lim_{\|h\|\to 0} \frac{\|f(u+h) - f(u) - \frac{\mathrm{d}f}{\mathrm{d}u}[u]h\|}{\|h\|} = 0 \tag{1.1}$$

where the notation $\|h\| \to 0$ is a shorthand for the requirement that the limit exist uniformly across sequences (h_n) in $\mathcal{U}$ such that $\|h_n\| \to 0$ as $n \to \infty$. It is important to observe that $\frac{\mathrm{d}f}{\mathrm{d}u}[u]$ is a linear operator that depends upon u, so that $\frac{\partial f}{\partial u}[u](v)$ is the Fréchet derivative at the location $u \in \mathcal{U}$ in the direction $v \in \mathcal{U}$. For a function $f : \mathcal{U}\times\mathcal{V} \to \mathcal{W}$ the partial Fréchet deriviative is defined analagously to be the operator $\frac{\partial f}{\partial u} \in \mathcal{L}(\mathcal{U}\times\mathcal{V},\mathcal{W})$ that satisfies

$$\lim_{\|h\|\to 0} \frac{\|f(u+h,v) - f(u,v) - \frac{\partial f}{\partial u}[u,v](h)\|}{\|h\|} = 0$$

whenever the above limit exists. Finally, consider the case where the function u depends on v. Let $f : \mathcal{U}\times\mathcal{V} \to \mathcal{W}$, suppose that $\mathcal{U}$ is a space of functions with domain $\mathcal{V}$. Then, when it exists, the Fréchet derivative of f with-respect-to v is the operator $\frac{\mathrm{d}f}{\mathrm{d}v} \in \mathcal{L}(\mathcal{U}\times\mathcal{V},\mathcal{W})$ that satisfies

$$\lim_{\|h\|\to 0} \frac{\|f(u(v+h),v+h) - f(u,v) - \frac{\mathrm{d}f}{\mathrm{d}v}[u(v),v](h)\|}{\|h\|} = 0. \tag{1.2}$$

This will sometimes be referred to as the *total* Fréchet derivative of f.

2. Background. In this section a formal presentation of local sensitivity analysis is provided. In subsection 2.1 the problem is introduced, while subsections 2.2 and 2.3 present forward and adjoint sensitivity analysis, respectively. Lastly in subsection 2.4 we will briefly discuss probabilistic numerical methods for the solution of PDEs, and discuss their similarity to this work.

2.1. Local Sensitivity Analysis. We begin by introducing the relevant spaces for the problem. Let $\mathcal{U}$, P, $\mathcal{F}$ and $\mathcal{G}$ each be real-valued Banach spaces. In this paper it will be assumed that $\mathcal{U}$ and $\mathcal{F}$ are infinite-dimensional spaces of functions defined on spatial domain D, with $\mathcal{U}$ referred to as the *solution space* and $\mathcal{F}$ as the *constraint space*. Define $\mathcal{U}_P$ to be a space of real-valued functions on $D \times P$ with the property that $u(\cdot, p) \in \mathcal{U}$ for all $p \in P$, and let $\mathcal{U}_{\partial P} = \left\{ \frac{\partial u}{\partial p} : u \in \mathcal{U}_P \right\}$. The set $\mathcal{F}_P$ is defined analogously. The *parameter space* P may be finite- or infinite-dimensional. The space $\mathcal{G}$ will be referred to as the *quantity of interest (QoI) space*, and will be assumed to

be finite-dimensional. In particular it will often be the case that $\dim(\mathcal{G}) = 1$, though we note that this is not required for the presentation below.

Two functions define the problem. The function $F : \mathcal{U} \times P \to \mathcal{F}$ is referred to as the *constraint function*, and loosely speaking this encapsulates all of the constraints that must be satisfied in order for a pair $(u, p) \in \mathcal{U} \times P$ to constitute a solution to the PDE. The function $g : \mathcal{U} \times P \to \mathcal{G}$ is referred to as the *QoI function*, and this describes a typically low-dimensional quantity of interest derived from the solution; in the context of optimisation problems this will generally be the objective function whose minimiser is sought.

More formally, F is such that for each $p \in P$ there is a unique $u^\dagger \in \mathcal{U}_P$ that satisfies $F(u^\dagger(\,\cdot\,, p), p) = 0$ for each $p \in P$. For convenience, define the *parameter-to-solution map* $U : P \to \mathcal{U}$ which provides the solution to the underlying differential equation for a particular value of the parameter, i.e. $U(p) = u^\dagger(\,\cdot\,, p)$. As a result, the equation $F(U(p), p) = 0$ is automatically satisfied for all $p \in P$.

It will be assumed the partial Fréchet derivatives of F and g with-respect-to both u and p exist and are tractably computable for all pairs $(u, p) \in \mathcal{U} \times P$. It will also be assumed that the derivative of U with-respect-to p exists but is not tractable. Note that this implies the existence of the total derivatives $\frac{\mathrm{d}F}{\mathrm{d}p}$ and $\frac{\mathrm{d}g}{\mathrm{d}p}$. Lastly we assume that $\frac{\partial F}{\partial u}[U(p), p]$ is nonsingular for each $p \in P$.

The objective is to estimate the value of the Fréchet derivative

$$\frac{\mathrm{d}g}{\mathrm{d}p}[U(p), p] \in \mathcal{L}(P, \mathcal{G})$$

for a pair $(U(p), p)$. Note that since the location at which the derivative is taken is $U(p)$, this should be interpreted as a total Fréchet derivative in the form of (1.2). To fix ideas we consider the following simple parameter sensitivity problem.

EXAMPLE 2.1 (Partial Differential Equation). *Let $P \subseteq \mathbb{R}^n$ be an open set. Consider the following parametrised steady state conductivity model:*

$$\begin{aligned} -\nabla \cdot (\kappa(x; p)\nabla u(x)) &= f(x) && x \in D \\ u(x) &= 0 && x \in \partial D \end{aligned}$$

where $f \in H^{-1}(D)$ and $\kappa : D \times P \to \mathbb{R}^{d \times d}$ satisfies $\lambda_p |e|^2 \leq e \cdot \kappa(x, p) e \leq \Lambda_p |e|^2$ for all $x \in D$ and $e \in \mathbb{R}^d$ for some $0 < \lambda_p < \Lambda_p < \infty$ for all $p \in P$. Standard existence theory for elliptic PDEs [17, Section 6.2, Theorem 3] states that a weak solution $u \in H^1_0(D)$ exists for every $p \in P$. For convenience we will suppose that the boundary conditions are implicitly satisfied, i.e. $\mathcal{U} = H^1_0(D)$. The constraint equation is given by $F(u, p) = -\nabla \cdot (\kappa(x; p)\nabla u(x)) - f(x)$ so that $\mathcal{F} = H^{-1}(D)$. Suppose that the quantity-of-interest is $g(x) = \|u\|_2 = \left(\int_D u^2(x)\, dx\right)^{\frac{1}{2}}$.

Both forward and adjoint sensitivities are computed by first observing that the total derivative of interest, $\frac{\mathrm{d}g}{\mathrm{d}p}$ satisfies the following identity:

$$\frac{\mathrm{d}g}{\mathrm{d}p}[U(p), p] = \frac{\partial g}{\partial u}[U(p), p]\frac{\mathrm{d}U}{\mathrm{d}p}[p] + \frac{\partial g}{\partial p}[U(p), p]. \tag{2.1}$$

Since it is assumed that $\frac{\partial g}{\partial u}$ and $\frac{\partial g}{\partial p}$ are each analytically tractable, the only remaining quantity that must be computed is $\frac{\mathrm{d}U}{\mathrm{d}p}$. The challenge is that since the parameter-to-solution map $U(p)$ is typically inaccessible and must be approximated independently for each $p \in P$, $\frac{\mathrm{d}U}{\mathrm{d}p}$ is also difficult to compute. The forward and adjoint approaches handle this intractability in different ways, which will now be presented.

2.2. Forward Sensitivity Analysis. In forward sensitivity analysis we seek to calculate $\frac{\mathrm{d}U}{\mathrm{d}p}$ directly. Note that we have

$$\frac{\partial F}{\partial p}[U(p),p] = \frac{\partial F}{\partial u}[U(p),p]\frac{\mathrm{d}U}{\mathrm{d}p}(p) + \frac{\partial F}{\partial p}[U(p),p], \quad p \in P.$$

Further, since by construction $F(U(p),p) = 0$, we also have that $\frac{\mathrm{d}F}{\partial p}[U(p),p] = 0$. This gives the forward sensitivity equation

$$\frac{\partial F}{\partial u}[U(p),p]\frac{\mathrm{d}U}{\mathrm{d}p}[U(p),p] = -\frac{\partial F}{\partial p}[U(p),p], \quad p \in P \tag{2.2}$$

which is a linear system whose solution can be computed to determined $\frac{\mathrm{d}U}{\mathrm{d}p}$, since $\frac{\partial F}{\partial u}$ is assumed to be invertible. This solution can then be substituted into (2.1) to compute $\frac{\mathrm{d}g}{\mathrm{d}p}$.

Note that both the operator $\frac{\partial F}{\partial u}[U(p),p]$ and the right-hand-side $-\frac{\partial F}{\partial p}[U(p),p]$ depend both on the parameter value p and the solution $U(p)$. This has two important consequences. Firstly, if sensitivities are required at another point $q \neq p$ then the solution $U(q)$ must be recomputed and the forward sensitivity equation (2.2) must be solved anew to determine $\frac{\mathrm{d}U}{\mathrm{d}p}[U(q),q]$. Secondly, for most problems of interest $U(p)$ will not be available explicitly and one must substitute an approximate solution $\hat{U}(p) \approx U(p)$. This may induce further numerical error, the impact of which must in turn be analysed, but also means that even though (2.2) is linear, its solution is unlikely to be available in closed-form owing to its dependence on $\hat{U}(p)$. We now consider the computation of the forward sensitivities for Example 2.1.

EXAMPLE 2.2 (Elliptic PDE: Forward Sensitivity Analysis). *We begin by deriving $\frac{\partial F}{\partial p}$. Assume that κ is once-differentiable in each coordinate of p and that $\sup_{x\in D}|\partial_{p_i}\kappa(x;p)| < \infty$. The Frechét derivative of F with respect to p at $(U(p),p)$ is defined by*

$$\frac{\partial F}{\partial p}[U(p),p]q = -\sum_{i=1}^{m} \nabla\cdot\left(\frac{\partial \kappa}{\partial p_i}(x;p)\nabla U(p)(x)\right) q_i, \quad q \in P. \tag{2.3}$$

From energy estimates for weak solutions of elliptic PDEs, $\nabla U(p)(x) \in L^2(D)$. For illustration, it is straightforward to show that the RHS of (2.3) lies in $H^{-1}(D)$. The derivative $\frac{\partial F}{\partial u}$ is given by

$$\frac{\partial F}{\partial u}[U(p),p](v) = -\nabla\cdot(\kappa(x;p)\nabla v(x)) \quad v \in \mathcal{U}. \tag{2.4}$$

so that clearly $\frac{\partial F}{\partial u}[U(p),p] \in \mathcal{F}$ since in this case, owing to the linearity of the PDE operator, $\frac{dF}{du}$ is identical to this operator and independent of both p and $U(p)$, though for general nonlinear problems this will not be the case. The sensitivities of U with respect to the p_i are therefore defined by the following system of PDEs

$$-\nabla\cdot\left(\kappa(x;p)\nabla\frac{dU(p)}{dp_i}(x)\right) = -\nabla\cdot\left(\frac{\partial\kappa}{\partial p_i}(x;p)\nabla U(p)(x)\right), \quad (x,p) \in D\times P. \tag{2.5}$$

For fixed p system of equations is well-posed, guaranteeing the existence of unique solutions $\frac{dU}{dp_i} \in H_0^1(D)$, $i = 1,\ldots,m$.

Once these m PDEs have been solved, the computed solutions can be substituted into (2.1) to determine $\frac{dg}{dp}$. To accomplish this we are required to compute the derivatives $\frac{\partial g}{\partial u}$ and $\frac{\partial g}{\partial p}$. Note that in this case g is independent of p, and it is further straightforward to show that

$$\frac{\partial g}{\partial u}[u](v) = \frac{dg}{du}[u](v) = \frac{\langle u, v\rangle_2}{\|u\|_2}.$$

Once again, note that this is a linear operator in v, but is nonlinear in u. We therefore have that

$$\frac{dg}{dp_i}[U(p), p] = \frac{1}{\|U(p)\|_2}\left\langle U(p), \frac{dU}{dp_i}\right\rangle_2$$

for the derivatives $\frac{dU}{dp_i}$ identified by solution of (2.5).

The central challenge with the forward approach, which motivates the adjoint approach that will be presented in the next section, is the dependence of the forward sensitivity equation (2.2) on the dimension of the parameter space: solving for $\frac{\mathrm{d}U}{\mathrm{d}p}$ requires the solution of $\dim(P)$ PDEs. In many practical problems the parameter space is extremely large; thus, a method for computing the sensitivities that is independent of the dimension of the parameter space is also of interest.

2.3. Adjoint Sensitivity Analysis. Adjoint sensitivity analysis begins by introducing the operator $\lambda \in \mathcal{L}(\mathcal{F}, \mathcal{G})$. Supposing that $\dim(\mathcal{G}) = n$, we can express g as $(g_1, \ldots, g_n)$ and consequently $\lambda = (\lambda_1, \ldots, \lambda_n)$ where $\lambda_i \in \mathcal{F}^*$ for $i = 1, \ldots, n$. For fixed p, the auxiliary term λ is selected to solve

$$\lambda_i \frac{\partial F}{\partial u}[U(p), p] = \frac{\partial g_i}{\partial u}[U(p), p], \quad i = 1, \ldots, n. \tag{2.6}$$

Assuming this is a unique solution λ exists, one can then recover the sensitivity of the quantity of interest g as follows

$$\frac{\mathrm{d}g_i}{\mathrm{d}p} = -\lambda_i \frac{\partial F}{\partial p} + \frac{\partial g}{\partial p}, \quad i = 1, \ldots, n. \tag{2.7}$$

which provides a computable expression for the local sensitivities.

We note that compared to subsection 2.2 which, in the finite-dimensional case, necessitates $m = \dim(P)$ solutions of the forward sensitivity equation, the adjoint system requires $n = \dim(G)$ solutions of the adjoint sensitivity equation. In typical situations where $n \ll m$ then there is a clear computational benefit to this approach.

EXAMPLE 2.3 (Elliptic PDE: Adjoint Sensitivity Analysis). *Recalling $\frac{\partial F}{\partial u}$ and $\frac{\partial g}{\partial u}$ as derived in Example 2.2, the problem that must be solved to identify $\lambda \in H_0^1(D)$ such that*

$$\nabla \cdot \left(\kappa^\top(x; p)\nabla\lambda(x)\right) = \frac{U(p)}{\|U(p)\|_2}. \tag{2.8}$$

Once λ has been determined, referring again to the derivation in Example 2.2 we have that

$$\frac{\partial g}{\partial p_i} = -\int \nabla\lambda(x) \cdot \frac{\partial \kappa}{\partial p_i}(x; p)\nabla U(p)(x)\, dx$$

which is real-valued, as required. Again note that in the equation that determines λ, $U(p)$ appears on the right-hand-side, so for each value of p for which sensitivities are required the PDE must be solved. Nevertheless the fact that in this example only a single system needs to be solved for each p makes the adjoint method significantly cheaper to apply when $m = dim(P)$ is large.

In the next section we will describe the new probabilistic approaches to both forward and adjoint sensitivity analysis, each of which operates with a constrained computational budget.

2.4. Probabilistic Numerical Methods for PDEs. When applied to PDEs, there is a marked similarity between this work and probabilistic numerical methods[2] applied to linear PDEs. In this section we will discuss these methods, and the similarity to the present approach. Broadly speaking these methods begin by placing a Gaussian prior on the function space occupied by the solution to the PDE. Finite-dimensional information about the unknown solution is then produced by projecting the linear PDE through a set of d functionals, referred to as *information functionals* in this work. The conjugacy of Gaussian distributions with linear projections can then be exploited to write down the posterior distribution in closed-form. For a detailed introduction to this perspective see [13, Chapter 5], in which it is referred to as the *probabilistic meshless method* (PMM).

This approach is equivalent to symmetric collocation with radial basis functions [55, 11], in that it is possible to construct the prior such that the posterior mean from PMM coincides with the estimator for the solution of the PDE produced in symmetric collocation. To our knowledge this approach was first presented in [55, Chapter 16], and extended in [11] to refine the error analysis, as well as explore applications in stochastic PDEs. In symmetric collocation the posterior distribution itself is not of interest, but the error analysis that appears in those works is relevant here as it provides an important interpretation for the posterior covariance. Specifically, the bound that appears in [55] connects the error to an object referred to as the *power function*, which can be shown to be directly connected to the posterior covariance that appears in the PMM.

In addition to the PMM, other works that could be interpreted as probabilistic numerical methods for PDEs include a series of papers that introduced *gamblets* for the solution of PDEs with rough coefficients [39, 41, 40]. These papers construct a probabilistic solution to the PDE in a broadly similar way to [13], but with several distinct differences. Firstly, the probability model is motivated by a game theoretic argument rather than Bayesian reasoning, though the ultimate conditioning procedure arrived at is equivalent. Secondly, the information about the solution is constructed in a distinctly different way, by projecting the defining equations of the PDE against a hierarchical basis formed by a nested partitioning of the domain, whereas in the PMM and in symmetric collocation it is obtained by evaluating those equations at a set of points referred to as *collocation points*. However this results in a very different error analysis, since collocation methods typically bound the estimation error in terms of the fill distance of these collocation points, whereas in gamblet-based methods, since there is no analogue of these points, a different approach must be adopted.

The chief similarities of these approaches to the approach presented in this paper is that, when the system defined by F is a PDE, the sensitivity equations will involve solving a system of PDEs. In this setting the approach that we describe is similar in

[2] See [25] for a high-level introduction, and [37] for a thorough literature review.

principal to the approaches we describe above, in that for a particular choice of prior and information, the method we employ will be equivalent to these methods. There are several distinct differences however. Firstly, it is possible that the system described by F is *not* a PDE, and indeed in this work we will explore sensitivity analysis for ODEs in addition to PDEs. While there exist probabilistic numerical methods for solving ODEs, they typically make approximations to account for nonlinearity which are not required in this work, as the systems which must be solved in sensitivity analysis are linear. Secondly, in the PDE case we do note make specific assumptions on the form of the information functionals, as these will typically be problem specific. Thirdly, by formulating the sensitivity equations as a single (degenerate) PDE on the joint space $D \times P$, the continuity of the sensitivities with respect to p is exploited to permit implicit interpolation of the sensitivities across different values of p. And lastly, the focus of this paper is on computing sensitivities, *not* on the solution of the PDE itself, which is assumed to be obtained by some classical numerical solver.

3. Probabilistic Approaches. In this section we will present two probabilistic approaches to computing parameter sensitivities. Each allows a user to restrict the amount of computational effort expended and still obtain an estimate of the sensitivities, while also providing an estimate of the error incurred as a result. Familiarity with Gaussian processes is assumed for this section; we refer the unfamiliar reader to the introduction given in [43]; see also [8] for a more mathematical treatment.

We will assume that there exist reproducing kernel Hilbert spaces (RKHSs) $\mathcal{U}'_P$, $\mathcal{F}'_P$ such that $\mathcal{U}'_P$ is dense in $\mathcal{U}_P$ and $\mathcal{F}'_P$ is dense in $\mathcal{F}_P$. Let $\mathcal{U}_{\partial P} = \left\{ \frac{\partial u}{\partial p} : u \in \mathcal{U}_P \right\}$ and let $\mathcal{U}'_{\partial P}$ be defined analogously for $\mathcal{U}'_P$. It will also be assumed that g is a functional, so that $\mathcal{G} = \mathbb{R}$; this last assumption can readily be generalised, and is made to simplify the presentation.

3.1. Probabilistic Forward Sensitivity Analysis. We first consider forward sensitivity analysis. We begin by modelling prior uncertainty about $\frac{\partial U}{\partial p}$ with the random variable X_F, distributed as $X_F \sim \mu_F = \mathcal{N}(a_F, C_F)$, where $a_F \in \mathcal{U}'_{\partial P}$ and $C_F : \mathcal{U}'_{\partial P} \to \mathcal{U}'_{\partial P}$ is a positive-definite covariance operator. It will be assumed that $\mu_F(\mathcal{U}_{\partial P}) = 1$. When $\dim(P) < \infty$ this prior takes the form of a vector-valued Gaussian process prior [1]. In the infinite-dimensional setting, we note that a discretisation of the parameter space will nevertheless be required for computational purposes, resulting in a parameter space that is effectively finite-dimensional, though a finite-dimensional parameter space is not strictly required for the theoretical results presented herein.

To obtain a posterior belief over the forward sensitivities, this prior will be conditioned on observations of (2.2). Let $\tilde{\mathcal{I}}_{F,1}, \ldots, \tilde{\mathcal{I}}_{F,d}$ be a such that $\tilde{\mathcal{I}}_{F,j} \in (\mathcal{F}^m)^*$ for $j = 1, \ldots, d$ and let $\{p_1, \ldots, p_d\} \subset P$. Let X_F be a random variable with law μ_F. Note that the prior distribution μ_F implies a prior distribution over $\frac{\mathrm{d}g}{\mathrm{d}p}$ by projecting through the linear map given in (2.1); this will be denoted ν_F. By applying each operator $\tilde{\mathcal{I}}_{F,j}$ to (2.2) we obtain

$$\tilde{\mathcal{I}}_{F,j} \frac{\partial F}{\partial u}[U(p_j), p_j] X_F = -\tilde{\mathcal{I}}_{F,j} \frac{\partial F}{\partial p}[U(p_j), p_j] \tag{3.1}$$

which, under the assumptions made at the start of this section, yields the information

$$f_{F,j} = -\tilde{\mathcal{I}}_{F,j} \frac{\partial F}{\partial p}[U(p_j), p_j]$$

where $f_{F,j} \in \mathbb{R}$. Let $f_F \in \mathbb{R}^d$ be the vector with $[f_F]_j = f_{F,j}$.

It is more mathematically convenient to think of the $\tilde{\mathcal{I}}_{F,j}$ in terms of functionals defined on $\mathcal{U}_{\partial P}$. To this end, let $\mathcal{I}_{F,j} \in \mathcal{U}_{\partial P}^*$ be defined by

$$\mathcal{I}_{F,j} \frac{\partial u}{\partial p} = \tilde{\mathcal{I}}_{F,j} \frac{\partial F}{\partial u}[U(p_j), p_j] \frac{\partial u}{\partial p}(\cdot, p_j).$$

We refer to $\mathcal{I}_{F,1}, \ldots, \mathcal{I}_{F,d}$ as the *information functionals*, and will assume that the information functionals are linearly independent.

The posterior is obtained by conditioning the prior on the information functionals. First, introduce the operator $\mathcal{I}_F : \mathcal{U}_P \to \mathbb{R}^d$, given by

$$\mathcal{I}_F \frac{\partial u}{\partial p} = \begin{bmatrix} \mathcal{I}_{F,1} \frac{\partial u}{\partial p} \\ \vdots \\ \mathcal{I}_{F,d} \frac{\partial u}{\partial p} \end{bmatrix}.$$

Then we seek to compute $X_F | \mathcal{I}_F X_F = f_F$. Owing to the linearity of $\mathcal{I}_F$, the resulting posterior distribution is again Gaussian and is given in the following proposition.

PROPOSITION 3.1 (Probabilistic Forward Sensitivity Analysis). *The posterior $X_F | \mathcal{I}_F X_F = f_F$ has law $\bar{\mu}_F$ given by*

$$\begin{aligned} \bar{\mu}_F &= \mathcal{N}(\bar{a}_F, \bar{C}_F) \\ \bar{a}_F &= a_F + C_F \mathcal{I}_F^\dagger [\mathcal{I}_F C_F \mathcal{I}_F^\dagger]^{-1} (f_F - \mathcal{I}_F a_F) \\ \bar{C}_F &= C_F - C_F \mathcal{I}_F^\dagger [\mathcal{I}_F C_F \mathcal{I}_F^\dagger]^{-1} \mathcal{I}_F C_F. \end{aligned}$$

The implied posterior distribution over $\frac{dg}{dp}$, denoted $\bar{\nu}_F$, is given by

$$\begin{aligned} \bar{\nu}_F &= \mathcal{N}(\bar{g}_F, \bar{G}_F) \\ \bar{g}_F(p) &= \frac{\partial g}{\partial u}[U(p), p](\bar{a}_F(\cdot, p)) + \frac{\partial g}{\partial p}[U(p), p] \\ \bar{G}_F(p, p') &= \frac{\partial g}{\partial u}[U(p), p] \delta[\cdot, p] \bar{C}_F \delta[\cdot, p']^\dagger \frac{\partial g}{\partial u}[U(p'), p']^\dagger. \end{aligned}$$

An important note is that even when underlying system described by F is nonlinear, the posterior distribution remains Gaussian owing to the linearity of the Fréchet derivatives. Choice of prior mean and covariance is highly problem specific, and will be discussed for the specific examples considered in this paper in section 5. Next we turn to the adjoint approach.

3.2. Probabilistic Adjoint Sensitivity Analysis. For the adjoint problem, the system that must be solved is now (2.6). Since $\mathcal{F}_P'$ is assumed to be an RKHS, due to the representer theorem (see e.g. [6, Section 4.4]) we have $\lambda f = \langle f, \beta \rangle_{\mathcal{F}}$, where $f, \beta \in \mathcal{F}_P'$,

The proposed approach is as in the previous section. We model uncertainty in β with the random variable X_A, whose law is $\mu_A = \mathcal{N}(a_A, C_A)$, where $a_A \in \mathcal{F}_P'$ and $C_A : \mathcal{F}_P' \to \mathcal{F}_P'$ is a positive-definite covariance operator. Note that this again implies a distribution ν_A over $\frac{\mathrm{d}g}{\mathrm{d}p}$ by projecting through the linear map

$$\mathcal{J}[p](\beta) = \left\langle \frac{\partial F}{\partial p}[U(p), p], \beta(\cdot, p) \right\rangle_{\mathcal{F}}.$$

An important remark, however, is that unless μ_A and μ_F are chosen carefully, the implied distributions ν_A and ν_F will not be equal.

To define the information functionals let $\{(e_1, p_1), \dots, (e_d, p_d)\} \subset \mathcal{U} \times P$. Then

$$\mathcal{I}_{A,j}\beta = \left\langle \frac{\partial F}{\partial u}[U(p_j), p_j](e_j), \beta(\cdot\,; p_j) \right\rangle_{\mathcal{F}}, \tag{3.2}$$

so that $\mathcal{I}_{A,j} \in \mathcal{F}_P^*$. Furthermore note that the information $f_{A,j} \coloneqq \frac{\partial g}{\partial u}[U(p_j), p_j](e_j)$ is clearly computable. Let f_A and $\mathcal{I}_A$ be defined analogously to previous sections; then the posterior on β is given in the following proposition.

PROPOSITION 3.2 (Probabilistic Adjoint Sensitivity Analysis). *The posterior distribution $\beta | f_A \sim \bar{\mu}_A$ is given by*

$$\begin{aligned}
\bar{\mu}_A &= \mathcal{N}(\bar{a}_A, \bar{C}_A) \\
\bar{a}_A &= a_A + C_A \mathcal{I}_A^\dagger (\mathcal{I}_A C_A \mathcal{I}_A^\dagger)^{-1} (f_A - \mathcal{I}_A a_A) \\
\bar{C}_A &= C_A - C_A \mathcal{I}_A^\dagger (\mathcal{I}_A C_A \mathcal{I}_A^\dagger)^{-1} \mathcal{I}_A C_A.
\end{aligned}$$

The implied posterior distribution $\bar{\nu}_A$ is given by

$$\begin{aligned}
\bar{\nu}_A &= \mathcal{N}(\bar{g}_A, \bar{G}_A) \\
\bar{g}_A(p) &= -\mathcal{J}[p](\bar{a}_A) + \frac{\partial g}{\partial p}[U(p), p] \\
\bar{G}_A(p, p') &= \mathcal{J}[p] C_A \mathcal{J}^\dagger[p']
\end{aligned}$$

Note that the form of the posterior over β is essentially identical to the form of the posterior from Proposition 3.1, modulo the choice of information functionals and prior. Next we will present some theoretical analysis of the forward and adjoint methods.

3.3. Theoretical Analysis. Our first theoretical result concerns a local error bound for the posterior mean in terms of the posterior covariance. This result is a general result about conditional distributions of Gaussian process, and so is not specific to either the forward or adjoint method; as a result we adopt generic notation.

PROPOSITION 3.3 (Local error bound). *Let $\mu = \mathcal{N}(a, C)$ be the prior, for $a \in \mathcal{H}_C$, and let $\bar{\mu} = \mathcal{N}(\bar{a}, \bar{C})$ be the posterior measure based on observations $\mathcal{I}u^\dagger = f$ where $u^\dagger \in \mathcal{H}_C$, $\mathcal{I} \in (\mathcal{H}_C^*)^d$ and $f \in \mathbb{R}^d$. Then we have that, for each $\mathcal{L} \in \mathcal{H}_C^*$*

$$|\mathcal{L}\bar{a} - \mathcal{L}u^\dagger| \leq (\mathcal{L}\bar{C}\mathcal{L}^\dagger)^{\frac{1}{2}} \|a - u^\dagger\|_{C^{-1}}.$$

The result from Proposition 3.3 is similar to results on error bounds in scattered data approximation with radial basis functions, such as in [55]. The term $(\mathcal{L}\bar{C}\mathcal{L}^\dagger)^{\frac{1}{2}}$ is analagous to the *power function* [55, Section 11.1], but the focus in that work is on the case when both $\mathcal{L}$ and $\mathcal{I}_j$ are evaluation functionals. In [55, Chapter 16] each of these restrictions is relaxed, however the form of the power function derived in this case is more abstract than presented here.

Similar bounds appear in the literature on solution of PDEs by symmetric collocation with radial basis functions (see e.g. [55, Section 16.3], [13, 11]). In these cases it is typically assumed that the $\tilde{\mathcal{I}}_j$ are evaluation functionals, so that the observations are point evaluations of the right-hand-side of the PDE, and that $\mathcal{L}$ is again an evaluation functional. It is then possible to bound $(\mathcal{L}\bar{C}\mathcal{L}^\dagger)^{\frac{1}{2}}$ in terms of the fill

distance in the interior and on the boundary of the domain. We have opted to make minimal assumptions on the form of the information operators and test functions in Proposition 3.3, to avoid tying the result to a particular numerical method. Further note that the cited results only apply for fixed p when performing sensitivity analysis for an elliptic PDE; as a global function of (x, p) the sensitivity analysis equations may not be elliptic even when for fixed p the underlying PDE is elliptic.

The next proposition provides theoretical guarantees for the setting when the solution $U(p)$ cannot be accessed directly, and instead a numerical estimate is provided by the map $\hat{U} : P \to \mathcal{U}$. The natural way to provide such guarantees is by bounding the distance between the measure conditioned based on $U(p)$ to that based on $\hat{U}(p)$. This is closely related to results that appear in [53, e.g. Theorem 4.6], though the results presented therein assume that the two measures have a common dominating measure, which is not the case in the present setting. A consequence of this is that the Hellinger metric, which is commonly used to measure distance in the space of probability measures in the field of uncertainty quantification, is not suitable here.

To proceed we introduce the 2-Wasserstein metric, which is suitable for measures that are mutually singular. Perhaps the most common way to define this metric is in terms of couplings of probability measures. Let μ_1 and μ_2 be measures on some abstract normed space $\mathcal{V}$. Let $\Gamma(\mu_1, \mu_2)$ be the set of couplings of μ_1 and μ_2, that is, the set of all Borel probability measures $\pi \in \mathcal{P}(\mathcal{V} \times \mathcal{V})$ with the property that $\pi(A \times \mathcal{V}) = \mu_1(A)$ and $\pi(\mathcal{V} \times A) = \mu_2(A)$ for each Borel set $A \subset \mathcal{V}$. Then the 2-Wasserstein metric [54, Definition 6.1] is given by

$$W_2(\mu_1, \mu_2) = \left(\inf_{\pi \in \Gamma(\mu_1, \mu_2)} \int_{\mathcal{V} \times \mathcal{V}} \|v - v'\|^2 \, \mathrm{d}\pi(v, v') \right)^{\frac{1}{2}}. \tag{3.3}$$

We now proceed to state a generic result concerning robustness to approximation error, which will then be applied to the methods described in Proposition 3.1 and Proposition 3.2.

PROPOSITION 3.4 (Robustness to Numerical Error). *Let $\mu = \mathcal{N}(a, C)$ be a Gaussian distribution with associated RKHS $\mathcal{H}$, for $a \in \mathcal{H}$ and $C : \mathcal{H} \to \mathcal{H}$ positive-definite. Assume that $\mathcal{I}$, $\hat{\mathcal{I}}$ are each bounded linear operators from $\mathcal{H}$ to $\mathbb{R}^d$. Let $\bar{\mu} = \mathcal{N}(\bar{a}, \bar{C})$ be the posterior measure based on observations $\mathcal{I}u^\dagger = f$ where $u^\dagger \in \mathcal{H}$. Let $\hat{\mu} = \mathcal{N}(\hat{a}, \hat{C})$ be the same prior conditioned on observations $\hat{\mathcal{I}}u^\dagger = \hat{f}$. Then it holds that*

$$W_2(\bar{\mu}, \hat{\mu}) \leq (C_{\mathcal{I},1} + C_{\mathcal{I},2}) \|\mathcal{I} - \hat{\mathcal{I}}\|_{\mathcal{H} \to \mathbb{R}^d} + C_f \|f - \hat{f}\|_{\mathbb{R}^d} + \mathcal{O}\left(\|\mathcal{I} - \hat{\mathcal{I}}\|^2_{\mathcal{H} \to \mathbb{R}^d} \right)$$

where

$$C_{\mathcal{I},1} = \|C\|_{\mathcal{H}} \left[\left(\|a\|_{\mathcal{H}} + \|C\|_{\mathsf{HS}}^{\frac{1}{2}} \right) \left(\|G^{-1}\mathcal{I}\|_{\mathcal{H} \to \mathbb{R}^d} + \|\hat{G}^{-1}\hat{\mathcal{I}}\|_{\mathcal{H} \to \mathbb{R}^d} \right) + \|G^{-1} f\|_{\mathbb{R}^d} \right]$$

$$C_{\mathcal{I},2} = \alpha \|G^{-1}\|^2_{\mathbb{R}^d} \|C\|_{\mathcal{H}} \left(\|G^{-1} f\|_{\mathbb{R}^d} + \|\mathcal{I}\|_{\mathcal{H} \to \mathbb{R}^d} \|\hat{\mathcal{I}}\|_{\mathcal{H} \to \mathbb{R}^d} \left(\|a\|_{\mathcal{H}} + \|C\|_{\mathsf{HS}}^{\frac{1}{2}} \right) \right)$$

$$C_f = \|C\|_{\mathcal{H}} \|G^{-1}\hat{\mathcal{I}}\|_{\mathcal{H} \to \mathbb{R}^d}$$

$$\alpha = \|\mathcal{I}C\|_{\mathcal{H} \to \mathbb{R}^d} + \|\hat{\mathcal{I}}C\|_{\mathcal{H} \to \mathbb{R}^d}$$

and $G = \mathcal{I}C\mathcal{I}^\dagger$, $\hat{G} = \hat{\mathcal{I}}C\hat{\mathcal{I}}^\dagger$.

We next prove a corollary of this result which establishes a bound for the error in the posterior distribution for both forward and adjoint sensitivity analysis as a result of the need to use $\hat{U}(p)$ rather than having access to $U(p)$ directly.

COROLLARY 3.5. *Assume that for each $p \in P$ there exists $\epsilon > 0$ such that*

$$\left\| \frac{\partial F}{\partial u}[U(p), p] - \frac{\partial F}{\partial u}[\hat{U}(p), p] \right\|_{\mathcal{U} \to \mathcal{F}} \leq \epsilon,$$

$$\left\| \frac{\partial F}{\partial p}[U(p), p] - \frac{\partial F}{\partial p}[\hat{U}(p), p] \right\|_{P \to \mathcal{F}} \leq \epsilon, \quad \textit{and}$$

$$\left\| \frac{\partial g}{\partial u}[U(p), p] - \frac{\partial g}{\partial u}[\hat{U}(p), p] \right\|_{\mathcal{U} \to \mathcal{G}} \leq \epsilon.$$

Further assume that the $\tilde{\mathcal{I}}_{F,j}$ and $\tilde{\mathcal{I}}_{A,j}$ are such that, for all $j = 1, \ldots, d$

$$\|\tilde{\mathcal{I}}_{F,j}\|_{\mathcal{F} \to \mathbb{R}} < M < \infty,$$

$$\|\tilde{\mathcal{I}}_{A,j}\|_{\mathcal{F} \to \mathbb{R}} < M < \infty.$$

Lastly assume that $\|\cdot\|_{\mathbb{R}^d} = \|\cdot\|_2$.

Let $\hat{\mu}_F$ be the posterior distribution from Proposition 3.1, with $\hat{U}(\cdot)$ substituted for $U(\cdot)$. Likewise let $\hat{\mu}_A$ be the posterior from Proposition 3.2 with the same substitution. Then we have

$$W_2(\bar{\mu}_F, \hat{\mu}_F) \leq (C_{\mathcal{I},1} + C_{\mathcal{I},2} + C_f) M \epsilon \sqrt{d} + \mathcal{O}(\epsilon^2)$$

$$W_2^2(\bar{\mu}_A, \hat{\mu}_A) \leq (C_{\mathcal{I},1} + C_{\mathcal{I},2} + C_f) M \epsilon \sqrt{d} + \mathcal{O}(\epsilon^2)$$

3.4. Comparison of Forward and Adjoint Approaches. We conclude this section with a brief discussion of the relative merits of the forward and adjoint approaches, compared to the classical approach.

Choice of Method. The forward approach requires the user to specify a prior on the parameter space; this is a space of dimension $\dim(P)$. While the space in which the prior is placed for the adjoint problem is less directly connected to the derivative of interest, which might make eliciting a prior more challenging, in the finite-dimensional case reasoning about the correlation structure between the components of $\frac{\partial u}{\partial p}$ for the forward problem may also be challenging. As a result, much as in classical sensitivity analysis, we are inclined to recommend the adjoint approach whenever $\dim(\mathcal{G}) < \dim(P)$, as will often be the case. However if the user has strong prior information about the correlation structure between these components, the forward approach may still perform well. Indeed, in the infinite-dimensional case such information is provided by knowledge about the smoothness of the function p.

Experimental Design. Propositions 3.1 and 3.2 each allow the user to construct a global model for the required derivatives. However in order to perform inference globally, one requires a set of points in P with which to construct the posterior. Both the forward and the adjoint approach suffer from the curse of dimensionality in this respect, since Gaussian processes typically require such designs to be "space-filling"[3], and if P is high-dimensional constructing a space filling design will be equally prohibitive in either mode. However in the present paper we focus on application of these methods within iterative optimisation algorithms, so that rather than requiring a space-filling design we only require good estimates of the gradient along the path in parameter space followed by the optimiser. This will be discussed in detail in the next section.

[3]Since, typically, the rate of convergence of Gaussian processes with this type of information depends on the "fill distance", i.e. the maximum distance of any point in the space to a design point. See e.g. [55], or [11] in the context of PDEs.

4. Optimisation and Probabilistic Sensitivity Analysis. We now explore a potential application of probabilistic local sensitivity analysis, as a way to provide approximations of gradients in optimisation algorithms. As a starting point we will consider the most fundamental of gradient-based optimisation algorithms, gradient descent [15]. In subsection 4.1 we briefly recall the GD algorithm. In subsection 4.2 we describe how probabilistic gradients can be incorporated into the algorithm. Then, in section 5 we explore the use of this approach in two applications.

4.1. Gradient Descent. We now describe the GD algorithm. GD is in many respects a prototypical gradient-based optimisation method, making it a natural starting point for studying the integration of probabilistic gradients into such algorithms. In GD the goal is to compute a (local) minimiser p^* of a function $g(p)$. To accomplish this a sequence of points (p^n), $p^n \in P, n \in \mathbb{N}$ is generated iteratively starting from some user-defined initial point p^0 and advancing according to

$$p^{n+1} = p^n - \gamma^n \frac{\mathrm{d}g}{\mathrm{d}p}(p^n)$$

where γ^n is a parameter of the method known as the *step size* or *learning rate*. Under specific conditions on f and γ^n it can be shown that $p^n \to p^*$ (again, a local minimiser) as $n \to \infty$; see [34, Section 3.2] for further details. GD is presented as an algorithm in Algorithm 1 in the supplement.

There are various methods for choosing the parameter γ_n. Since the focus of this work is on the performance when $\frac{\mathrm{d}g}{\mathrm{d}p}$ is replaced by the probabilistic gradients introduced in section 3, we will use a probabilistic version of the backtracking line search method described in [34, Algorithm 3.1], based on the method described in [32].

4.2. Gradient Descent with Probabilistic Gradients. We now discuss a probabilistic modification of GD. Heuristically the approach followed is to replace the computation of $\frac{\mathrm{d}g}{\mathrm{d}p}$ with a probabilistic gradient obtained from either Proposition 3.1 or Proposition 3.2; to simplify the exposition we will describe the former, but the approach is essentially identical in the latter. The approach is presented as an algorithm in Algorithm 4.1. Essentially, we begin with a prior μ_F which is projected to ν_F as described in Proposition 3.1. We then construct a sequence of random variables (X_F^n), where X_F^0 has law ν_F, by sequentially updating this prior with information collected over the course of the optimisation. This provides a posterior distribution over the gradient which is used in place of $\frac{\partial g}{\partial p}$ in GD. The principal advantages, illustrated in section 5, are that (i) for each value of p^n, one can often obtain an approximation of $\frac{\mathrm{d}g}{\mathrm{d}p}$ that is sufficiently accurate for the purposes of taking a gradient step, at a lower cost than that of computing $\frac{\mathrm{d}g}{\mathrm{d}p}$ directly, and (ii) since the posterior is defined over the entire parameter space, for some values of p^n *no* inversion problem must be solved to advance the gradient descent.

There are two main issues to address. The first is that that it is well-established in the literature on stochastic gradient descent that line-search algorithms such as the BLS routine are not robust to inaccurate gradients. This is discussed in [32]. Since the gradients we propose to use in this work are also inaccurate, an alternative line-search strategy for selecting the step sizes γ^n must be adopted in the probabilistic case. Borrowing from the literature on stochastic gradient descent, our proposed approach incorporates ideas from the probabilistic line search of [32] into the backtracking line search from [34, Section 3.2]. The PLS routine is described in Algorithm 4.2.

A second issue is that if X_F^n is not sufficiently accurate, the step size γ^n found by the probabilistic line search will be selected to be below the tolerance ϵ, causing the algorithm to terminate. To address this we propose to couple the computation of γ^n with the calculation of the gradient, as described in PROBJAC within Algorithm 4.1. Once the tolerance has been achieved, we calculate the step size γ according to a probabilistic version of backtracking line search that will be described presently. If γ is above the tolerance the procedure returns the current gradient estimate, along with the posterior distribution and the step size; otherwise, the tolerance δ is reduced and the conditioning procedure is repeated. This continues until delta is below some minimum value $\delta_{\min}$, at which point convergence is accepted.

Algorithm 4.1 Probabilistic version of gradient descent. The routines METRIC and INFO are problem specific and must be supplied by the user, with the former assessing the distribution of the currently computed posterior distribution to determine whether it is sufficiently narrow to accept it as a valid gradient and the latter supplying information, iteratively, based on the current distribution and location. The routine CONDITION implements Proposition 3.1. PLS is the probabilistic version of the Armijo line search, and is given in Algorithm 4.2. Of the new parameters, δ reflects how much accuracy is demanded of the posterior at each iteration, $\delta_{\min}$ specifies a maximum level of accuracy to protect against numerical instabilities resulting from large Gram matrices in CONDITION, and τ describes how rapidly δ is reduced when a valid descent direction cannot be found.

procedure PGD($p^0, g, \mu_F^0, \epsilon, \delta, \delta_{\min}, \tau_1$)
  Compute ν_F^0 from μ_F^0 and let X_F^0 be the random variable with law ν_F^0
  for $n = 1, 2, \ldots$ **do**
    $s_F^n, X_F^n, \gamma_n \leftarrow$ PROBJAC($X_F^{n-1}, g, p^{n-1}, \epsilon, \delta, \delta_{\min}$)
    if $\gamma^n < \epsilon$ **then**
      return p^{n-1}
    end if
    $p^n \leftarrow p^{n-1} + \gamma^n s_F^n$
  end for
end procedure
procedure PROBJAC($X, g, p, \epsilon, \delta, \delta_{\min}, \tau_1$)
  while $\delta > \delta_{\min}$ **do**
    while METRIC(X) $> \delta$ **do**
      $\mathcal{I}, f \leftarrow$ INFO(X, p)
      $X \leftarrow$ CONDITION($X, \mathcal{I}, f$)
      $s \leftarrow -\mathbb{E}(X(p))/\|\mathbb{E}(X(p))\|_2$
      $\gamma \leftarrow$ PLS(p, g, X)
      if $\gamma < \epsilon$ **then**
        $\delta \leftarrow \tau_1 \delta$
      else
        return s, X, γ
      end if
    end while
  end while
end procedure

Algorithm 4.2 Probabilistic line search algorithm. This is essentially a modification of the backtracking line search described in [34, Algorithm 3.1] to account for the fact that the gradient is a random variable rather than a constant. The parameters p, g and X are the parameter value, objective function and current posterior, respectively. The remaining parameters control the behaviour of the algorithm; we have specified sensible defaults for these and assume those defaults are used throughout the text. τ_2 controls how rapidly γ is decreased, while c controls how large a reduction in the objective function is required when a step is taken in the chosen direction and P^{crit} is the probability with which this reduction must be achieved. γ and $\gamma^{\min}$ control the initial and minimum values of γ respectively.

procedure PLS($p, g, X;\ \tau_2 = 0.5, c = 0.5, P^{\text{crit}}, \gamma = 1, \gamma^{\min} = 10^{-10}$)
  $s \leftarrow -\mathbb{E}(X)/\|\mathbb{E}(X)\|_2$
  while $\gamma > \gamma^{\min}$ **do**
    $p_\gamma \leftarrow p + \gamma s$
    if $g(p_\gamma) > g(p)$ **then**
      continue
    end if
    $Z \leftarrow -c\gamma X^\top s$
    if $\mathbb{P}(Z > g(p_\gamma) - g(p)) < P^{\text{crit}}$ **then**
      return γ
    end if
    $\gamma \leftarrow \tau_2 \gamma$
  end while
end procedure

4.2.1. Discussion. We now provide some important remarks about the algorithm presented above.

Choice of Direction. The direction chosen in Algorithm 4.1 at each iteration is the posterior mean. A natural alternative would be to instead *sample* a direction from the posterior distribution. This requires only minor modification of the above algorithm, but empirically was found to perform slightly worse in general; consequently we have opted to use the posterior mean as the descent direction.

Recycling Information. Note that the gradient here is computed based on information collected at all points $p_F^1, \ldots, p_F^n$, i.e. based on a *global* model for the gradient as a function of p. Since the sequence (p_F^n) will increasingly concentrate in a region of p_* as n increases, one expects that the prior $\bar{\mu}_F^{n-1}$ will be an increasingly accurate predictor for the gradient $\frac{\mathrm{d}g}{\mathrm{d}p}(p_F^n)$ as n increases. This means that once some computational effort has been expended to obtain a relatively accurate gradient, it is possible for PROBJAC to perform many further iterations based on this gradient without needing calls to CONDITION, as we shall see in section 5.

Linearly Independent Information. A global model introduces some additional burden to ensure that $\mathcal{I}_F^n$ is linearly independent of $\mathcal{I}_F^1, \ldots, \mathcal{I}_F^{n-1}$, both to maximise the amount of new information obtained at each p^n and to ensure that the linear system that must be solved to compute the posterior does not become singular. Thus, INFO must be carefully designed to ensure that the information returned is not too highly correlated with information already observed.

Computational Cost. To compute the posterior distributions from Proposition 3.1 and Proposition 3.2, it is necessary to compute the matrix $M = (\mathcal{I}C\mathcal{I}^\dagger)^{-1}\mathcal{I}C$ by solv-

ing the linear system $\mathcal{I}C\mathcal{I}^\dagger M = \mathcal{I}C$. To accomplish this one typically computes a Cholesky factorisation of $\mathcal{I}C\mathcal{I}^\dagger$, which becomes computationally intensive once many information functionals have been collected. However, we note that the sequential nature of the algorithm proposed is such that, rather than recomputing the full factorisation at each iteration of PROBJAC, one can use an updating formula for the factorisation such as presented in [38, Appendix B]; this is described in detail in section S2. In brief, one must only compute the Cholesky factorisation of a smaller matrix, whose dimension is only the same size as the dimension of the *new* information, which naturally dramatically reduces the cost of computing the probabilistic gradients.

The other factor that influences the cost is the size of $\mathcal{I}C$, and since this defines *how many* linear systems must be solved, it may be that ultimately the cost of assembling the posterior $\bar{\mu}_F^n$ exceeds than that of simply computing $\frac{\mathrm{d}g}{\mathrm{d}p}(p_F^n)$ despite the efficient updating formula for the factorisation. Thus in practise we propose that the PROBJAC is used only to perform the initial iterations, and that when the method is determined to be close to the truth, or the cost of constructing the posterior is too great, we revert to classical GD to complete the optimisation. In section 5 we adopt the crude rule of thumb that PROBJAC is terminated when the dimension of $\mathcal{I}C\mathcal{I}^\dagger$ exceeds $10,000$, though this is never exceeded in practise for one of the two examples examined. In future work more sophisticated switching schemes will be explored.

Choice of Metric. The routine METRIC must assess whether the posterior distribution at a particular iteration is sufficiently accurate for the probabilistic gradient to be accepted as a valid direction for the gradient descent. To determine this we focus on the width of the posterior covariance, and in this work we exclusively use the square-root of the trace of the posterior covariance, $\sqrt{\mathrm{trace}(\bar{G}_F)}$ as a proxy for the width. An exploration of other choices is not expected to affect the performance of the algorithm dramatically, and is reserved for future work.

Choice of Information Functionals. Lastly, we note that we have not yet discussed the selection of information functionals in INFO. We expect this to be highly problem dependent. We make a proposal in the next section that appears to be well adapted to the two examples presented therein, but do not expect that there exists a unique optimal choice of information for all settings.

5. Applications.

In this section we apply Algorithm 4.1 to compute the maximum *a-posteriori* (MAP) point in Bayesian inversion problems for two problems. In subsection 5.1 we seek to infer a small number of parameters of an ODE using the forward approach, and in subsection 5.2 inference of a larger number of parameters of a challenging PDE using the adjoint approach.

5.1. FitzHugh—Nagumo Model.

As a first example we examine the problem of inferring the parameters for the Fitzhugh—Nagumo model [19], a nonlinear oscillatory ODE. Since this problem has four parameters, we use the forward approach from subsection 3.1.

5.1.1. Problem Definition.

The equations that define the FitzHugh—Nagumo model are

$$\frac{\mathrm{d}v}{\mathrm{d}t} = v - \frac{v^3}{3} - w + I \qquad \frac{\mathrm{d}w}{\mathrm{d}t} = \frac{v + a - bw}{\tau}$$

where $a, b, I, \tau \in \mathbb{R}_+$ are parameters of the model. We concatenate the parameters as $p = [I, a, b, \tau]^\top \in \mathbb{R}^4 = P$. The solution to this system of ODEs for

$p^* = [0.5, 0.8, 0.7, 12.5]^\top$ is shown in the supplement in Figure S1a, while sensitivities are displayed in Figures S1b and S1c. The solution space $\mathcal{U}$ is a space of once differentiable functions $u : D \to \mathbb{R}^2$, where $D = [0, T]$ for some $T > 0$. A reformulation of this problem in terms of the constraint function $F(u, p)$ can be found in section S4, along with the form of its derivatives $\frac{\partial F}{\partial u}$, $\frac{\partial F}{\partial p}$.

To set up the inference problem we generated data for true parameter values p^* by evaluating the $v(t_i^{\text{data}}; p^*)$ at times $t_i^{\text{data}} = i$, $i = 1, \ldots, 20$. These locations are distinguished as dashed gray lines in Figure S1 in the supplement. Observations were then corrupted with centred Gaussian noise with standard deviation 10^{-2}, i.e. $y_i = v(t_i^{\text{data}}; p^*) + \xi_i$ where $\xi_i \sim \mathcal{N}(0, 10^{-2})$ IID. The prior distribution over the parameters was set to be log-Gaussian with mean $m_p = [1, 1, 1, 10]^\top$ and covariance I. The objective function is twice the negative logarithm of the likelihood multiplied by the prior, and is thus given by

$$g(p) := \frac{1}{\gamma^2} \sum_{i=1}^{M} (v(t_i^{\text{data}}; p) - y_i)^2 + (\log(p) - \mu)^\top \Sigma^{-1} (\log(p) - \mu)$$

5.1.2. Probabilistic Gradient Descent. To apply the probabilistic gradient descent algorithm from Algorithm 4.1 we must first specify the prior over $\frac{\mathrm{d}U}{\mathrm{d}p}$. Since the parameter space is four-dimensional and $\mathcal{U}$ is a space of vector-valued functions, formally $\frac{\mathrm{d}U}{\mathrm{d}p}$ is $\mathbb{R}^{2\times 4}$-valued. For convenience, we place a prior on $X : D \times P \to \mathbb{R}^8$, and form $\frac{\mathrm{d}U}{\mathrm{d}p}$ as

$$\frac{\mathrm{d}U}{\mathrm{d}p} = \begin{bmatrix} X_{1:4}^\top \\ X_{5:8}^\top \end{bmatrix}$$

where $X_{i:j}$ denotes components i to j of X. Noting that the posterior covariance is independent of the data, we assume an independent and identical prior over each column of $\frac{\mathrm{d}U}{\mathrm{d}p}$, so that the inference is identical but for the distinct right-hand-side for each component of p in the posterior mean of Proposition 3.1.

Since the initial condition is independent of p, this prior was taken to be $X \sim \mathcal{N}(\mathbf{0}, k)$ where

$$k((t, p), (t', p')) = Cq(t)q(t')k_{5/2}([t, p]^\top, [t', p']^\top \sigma, L)$$

$$k_{5/2}(r, r'; \sigma, L) = \sigma^2 \left(1 + \sqrt{5} d(r, r'; L) + \frac{5}{3} d(r, r'; L)^2\right) \exp\left(-\sqrt{5} d(r, r'; L)\right) \tag{5.1}$$

$$d(r, r'; L) = \sqrt{r^\top L^{-1} r'}$$

$$C = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

$$q(t) = t.$$

Multiplication by the linear functions $q(t)$ ensures that there is no uncertainty at $t = 0$, where the sensitivity is known to be zero.

The kernel $k_{5/2}$ in (5.1) is a member of the Matérn family [43, Section 4.2] and is the covariance kernel for a prior over functions with at least two continuous derivatives. To ease computation the length-scale matrix L was selected to be diagonal, $L = \text{diag}(\ell)$ for $\ell \in \mathbb{R}^6$. This parameter was further restricted to $\ell = [\ell_x \mathbf{1}_2, \ell_p \mathbf{1}_4]$ where $\ell_x, \ell_p \in \mathbb{R}$. The scalars σ, ℓ_x and ℓ_p were then selected by maximising the marginal likelihood of an initial candidate design (see e.g. [43, Section 5.4]). This was obtained

by sampling a set of candidate parameters p_i^{calib}, $i = 1, \dots, 5$ from the prior over the parameters and defining the corresponding evaluation functionals $\tilde{I}_{ij} = \delta[i]$, $i = 1, \dots, 20$ (i.e. using equally spaced points inside the spatial domain). The parameter ρ, which describes the degree of prior covariance between the components $u_1 = v$ and $u_2 = w$, was fixed to 0.5.

For this problem it was convenient to restrict the information functionals to be evaluation functionals, i.e. $\tilde{\mathcal{I}}_i = \delta[t_i^{\text{info}}]$. The points t_i^{info} were restricted to a fine grid of 1000 points in $(0, T]$, denoted $t_1^{\text{info}}, \dots, t_{1000}^{\text{info}}$. To choose the next information functionals at iteration n within the function INFO in Algorithm 4.1, we choose new conditioning locations within this set by attempting to minimise a heuristic based on the fill distance which often appears as an upper bound in Gaussian process regression problems. To be specific, we begin by constructing an augmented point set:

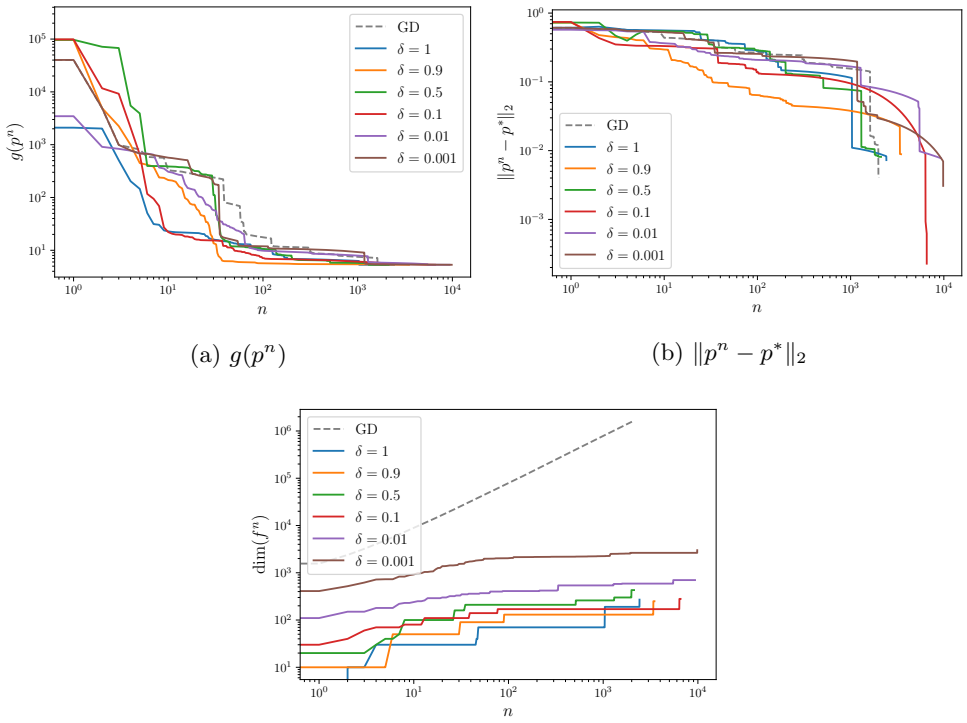
$$z_{ij} = \begin{bmatrix} t_i^{\text{info}} \\ p^j \end{bmatrix}.$$

for $j = 1, \dots, n$ denoting the iteration number in PROBJAC and p^j the corresponding parameter value for that iteration. The information functionals were then selected to be the $\tilde{\mathcal{I}}_j$ for which the distance between z_{in} and $z_{i'j}$, $i, i' = 1, \dots, 1000$, $j = 1, \dots, n-1$, is maximised.

5.1.3. Results. The paths taken by the probabilistic optimiser are contrasted with classical gradient descent in Figure 5.1. Figure 5.1a shows the value of $g(p^n)$, while Figure 5.1b shows the distance from the minimum obtained by gradient descent. All of the methods were started from the initial parameter value $p_0 = m_p$, and the GD tolerance was set to $\epsilon = 10^{-6}$. The threshold δ was varied from 1 (representing a high level of allowed error in the posterior gradient estimate) to 0.001 (representing a low level of allowed error). In each case $\delta_{\min}$ was set to 10^{-6}. For $\delta = 0.9, 0.5, 0.1$ the performance of the probabilistic approach is initially worse, as expected, though as the iterates near p^* the performance of the probabilistic approaches improves. Interestingly, for $\delta = 1$ and $\delta = 0.01$ the probabilistic approach actually seems to initially converge *faster* than the classical approach. This should not generally be expected, though we note that since the GD directions have no particular optimality properties nothing prevents an approximate method from achieving faster convergence.

Figure 5.1c tracks the amount of data collected (i.e. the size of f_F^n) as a function of the iteration number. This exhibits the expected behaviour of increasing inversely proportional to δ. However it is noteworthy that even in the strictest case, $\delta = 0.001$, only 3000 evaluations of $\frac{\partial F}{\partial p}$ are required over the course of 9840 iterations to perform almost as well as as gradient descent. For context computing the gradient $\frac{\mathrm{d}g}{\mathrm{d}p}$ using the `DOP853` algorithm [23, Section II] method as implemented in `scipy` required an average of 781 evaluations of $\frac{\partial F}{\partial p}$ *per iteration* of gradient descent, with a total of over 1.5 million evaluations over the course of the 2013 iterations performed with exact gradients. While $\frac{\partial F}{\partial p}$ is cheap to evaluate in this example, in a setting in which this was a bottleneck it is clear that the probabilistic method would be preferable. Further note that while 3000 evaluations of $\frac{\partial F}{\partial p}$ were required, as noted in subsection 4.2.1 this does not translate directly to inversion of a 3000×3000 Gram matrix, as the updating formula for Cholesky factorisations was exploited.

5.2. Groundwater Flow Model. We now consider a linear PDE that describes the steady-state flow of fluid through a porous medium. In this section the parameter is formally function-valued. Since after discretisation its dimension can be large, the



(a) $g(p^n)$

(b) $\|p^n - p^*\|_2$

(c) $\dim(f^n)$

Fig. 5.1: Performance of the probabilistic gradient descent algorithm on the FitzHugh—Nagumo model described in subsection 5.1 as the parameter δ, which roughly controlling the accuracy demanded of the probabilistic gradient estimate, is varied. Here n is the iteration number. Figure 5.1c shows the value of the quantity of interest g, in this case the value of the negative log-target in a Bayesian inference problem described in subsection 5.1. Figure 5.1b shows the distance from the parameter at iteration n to the true MAP point. Figure 5.1c shows the dimension of the matrix inversion problem that was solved in order to compute the posterior distribution.

adjoint approach is adopted.

5.2.1. Problem Definition. For a fixed value of the parameter p, the forward model is given by

$$\begin{aligned} -\nabla \cdot (p(x)\nabla u(x)) &= 0 && x \in D \\ u(x) &= x && x_2 = 0 \\ u(x) &= 1 - x && x_2 = 1 \\ \frac{\partial u}{\partial x_1} &= 0 && x_1 = 0 \text{ or } 1 \end{aligned}$$

Here the domain $D = [0,1]^2$ and $p : D \to \mathbb{R}$. We assume that $p(x) > 0$ for all $x \in D$.

The solution $u(x)$ was obtained by discretising the domain above with FEM on

a fine triangular meshing of the unit square based on a grid of 32×32 points using piecewise-linear basis functions. The mesh is depicted in Figure S2a in the supplement, and the discretisation results in a finite-dimensional approximation of the solution $u(x)$ in with 1089 degrees of freedom. The solution to the PDE above for the parameter value $p = 1$ is depicted in Figure S2b, again found in the supplement.

The parameter is defined to be piecewise constant over supersets of the cells of this mesh, defined by grouping the cells based on a subdivision of the domain into squares. For a parameter $N > 1$ these are obtained by placing down a regular grid of N^2 points, with $N + 1$ equispaced points along each axis. The points of this grid form the vertices of the N^2 parameter cells. In Figure S2a, the parameter cells for $N = 4$ are surrounded by green lines.

To construct the inverse problem, we use a Gaussian prior $p \sim \mathcal{N}(\mu, \Sigma) =: \pi(p)$ with $\mu = 5\mathbf{1}_{N^2}$, where $\mathbf{1}_{N^2}$ here denotes the vector of ones in $\mathbb{R}^{N^2}$. Letting x_i^{param} denote the centroid of cell i according to some arbitrary ordering of the cells, $i = 1, \dots, N$, the covariance is given by $\Sigma_{ij} = k(x_i^{\text{param}}, x_j^{\text{param}})$, where k is the Matérn 5/2 kernel given in (5.1), with amplitude and length-scale each set to 1. The data-generating parameter p^* was sampled from the prior over p. To define the likelihood, data was obtained by taking direct measurements of the solution $u(x; p^*)$ at locations $x_1^{\text{data}}, \dots x_M^{\text{data}}$ where $M = 25$ the x_j^{data} are the nearest mesh points to points on a regular 5×5 grid starting at $(0.1, 0.1)$ and ending at $(0.9, 0.9)$. The points of this grid are shown in Figure S2b in the supplement. Let $\tilde{y} \in \mathbb{R}^M$ be the vector with $\tilde{y}_j = u(x_j^{\text{data}}; p^*)$. These points were corrupted with IID Gaussian noise $\xi_j \sim \mathcal{N}(0, \gamma)$, $\gamma = 0.01$ $j = 1, \dots, M$ to obtain data $y = \tilde{y} + \xi$. Denoting the likelihood by $\pi(p|y, u)$ with dependence on u emphasised, the QoI for gradient descent was then given by $g(u, p) = -2 \log \pi(p|y, u)\pi(p)$, i.e.

$$g(p) := \frac{1}{\gamma^2} \sum_{i=1}^{M} (u(x_i^{\text{data}}; p) - y_i)^2 + (p - \mu)^\top \Sigma^{-1} (p - \mu) \tag{5.2}$$

5.2.2. Probabilistic Gradient Descent. To test the algorithm described in subsection 4.2 we attempt to compute the MAP point of the posterior distribution for the inverse problem described above. Owing to the potentially high dimension of the problem to be solved, the adjoint approach was used. For the prior we used $\beta \sim \mathcal{N}(0, k)$, where k is given by

$$k((x, p), (x', p')) = k_{52}((x, p), (x', p'); \sigma, \ell) q(x_2) q(x_2').$$

Here $q(x) = 1 - (2x - 1)^2$, so that $q(0) = q(1) = 0$, ensuring that the relevant boundary condition is encoded in the prior since we note that the boundary conditions do not depend upon p. Thus, the prior is formally over functions from $\mathbb{R}^{N^2+2}$ to $\mathbb{R}$, though since the problem has been discretised with the finite-element method the discretised prior is finite-dimensional. Strictly speaking to project the prior into the finite-element space requires computing integrals of the form $\int k(x, x')\phi_j(x)\mathrm{d}x$ for $j = 1, \dots, 1089$, however since these integrals do not generally have a closed-form we opt to approximate them as $\int k(x, x')\phi_j(x)\mathrm{d}x \approx k(x_j, x')$ where x_j is the nodal point corresponding to the basis function ϕ_j.

For the parameters of the prior, a separate constant length-scale was assigned to the spatial variables and the parameters, denoted ℓ_x and ℓ_p respectively, i.e. $\ell = [\ell_x \mathbf{1}_2, \ell_p \mathbf{1}_{N^2}]$. The amplitude σ and the length-scale ℓ_p were again selected by maximising the marginal likelihood of these parameters given a candidate design obtained again by sampling a set of candidate parameters p_i^{calib}, $i = 1, \dots, 10$,

from the prior over parameters, and choosing corresponding information functionals $\tilde{\mathcal{I}}_{ij}u = \int_D u(x)\phi_j(x)\mathrm{d}x$. Here the ϕ_j are the finite element basis functions corresponding to the nearest mesh points to a regular 10×10 grid of points within D, with basis functions on the top and bottom boundaries excluded.

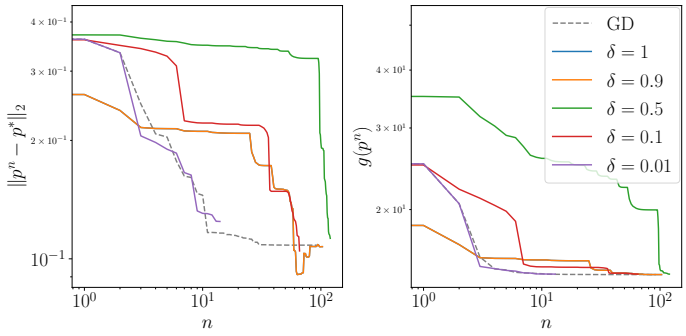
For the remaining parameter, ℓ_x, we note that since in (5.2) g depends only on the value of u at the points x_i^{data}, we therefore have that $\frac{\partial g}{\partial u}$ is zero everywhere but at these locations. Since this function is so rough, it is impossible to infer the spatial length-scale ℓ_x from evaluations of it. As a result, we opted to fix $\ell_x = 0.2$, based on the observed smoothness of the solution to the adjoint equations.

For the information functionals we selected $\tilde{\mathcal{I}}_j u = \int u(x)\phi_j(x)\mathrm{d}x$, i.e. projection against the j^{th} finite element basis function. This is straightforward to implement since after discretisation it is simply projection against the canonical basis vector $\boldsymbol{e}_j^\top$. The function INFO was implemented similarly to in subsection 5.1, with the fine grid of points now consisting of the mesh locations which the basis functions correspond to, again excluding points on the top and bottom boundaries. However, to ensure that the information f is nonzero, we enforce that when $\text{METRIC}(X^n) > \delta$, the first locations to be conditioned upon are those basis functions corresponding to x_i^{data}.

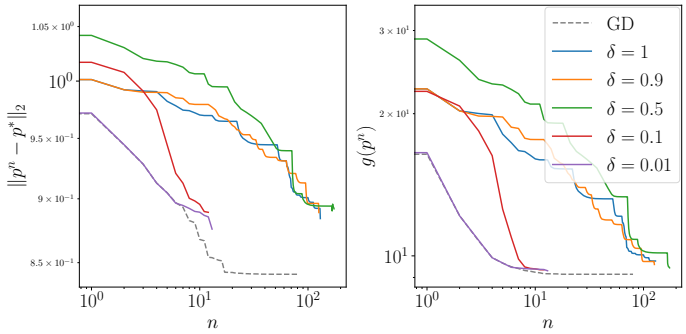
5.2.3. Results. The results of the optimisation are displayed in Figure 5.2. As in subsection 5.1 one can clearly see the behaviour of the method reverting to that of gradient descent as the size of δ is decreased. Further, performance appears to be broadly similar as the parameter dimension increases, reflecting that only a single function $\beta(x)$ must be learned, rather than $\frac{\mathrm{d}U}{\mathrm{d}p_i}$ for $i = 1, \ldots, \dim(P)$ as would be required in the forward approach. Thus, the output dimension of the inferred function is independent of the parameter dimension. While the *input* dimension does grow with $\dim(P)$, for the purposes of the gradient descent algorithm, at iteration n only the quality of inferences at and in the region of p^n is relevant. Since these points concentrate near p^*, performance does not appear to decay as the input dimension grows.

Figure 5.3 compares the cost of the probabilistic approach with that of the classical approach, for $N = 2$, $\dim(P) = 4$, by plotting the size of the matrix whose Cholesky factorisation that must be computed at each iteration in order to update the Cholesky factorisation of the Gram matrix with novel information, as discussed in subsection 4.2.1. We note that in general more information seems to be required than for the Fitzhugh-Nagumo example, so that the $10,000 \times 10,000$ limit on the size of the Gram matrix discussed in subsection 4.2.1 is generally what causes the algorithm to terminate, though from Figure 5.2 it is clear that nevertheless PROBJAC is *close* to convergence when this occurs. The higher cost is perhaps due to the fact that the right-hand side, $\frac{\partial g}{\partial u}$, is highly localised in this example. It is nevertheless the case that throughout the gradient descent procedure, the size of the inversion problem that must be computed with the probabilistic approach is significantly smaller than that which must be computed with the classical approach, though since the matrix inverted in the classical approach is sparse the costs are not directly comparable. Furthermore, as in subsection 5.1, for larger values of δ the approach shows the behaviour of being able to conduct a large number of iterations without needing to collect *any* evaluations of the right-hand-side, due to the fact that the model is global over parameter space.

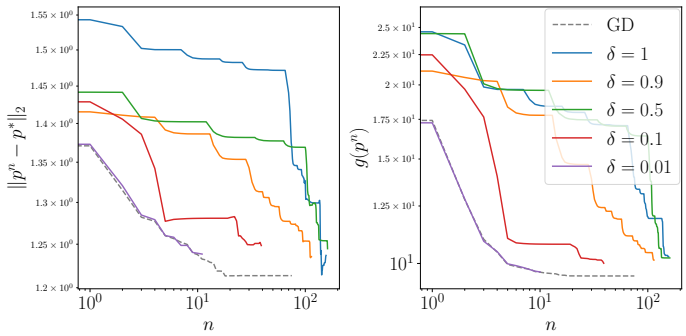
6. Conclusion. In this paper we have presented a probabilistic approach to computing local sensitivities of differential equation models in both the forward and adjoint modes. We presented an approach for incorporating these probabilistic gradients into a gradient descent algorithm, and examined the properties of this algorithm



(a) $N = 2$; $\dim(P) = 4$



(b) $N = 4$; $\dim(P) = 16$



(c) $N = 8$; $\dim(P) = 64$

Fig. 5.2: Results for the groundwater flow example from subsection 5.2, for a variety of parameter dimensions. In each row, the left-hand plot shows the distance between the parameter found at iteration n and the true value p^* of the parameter. The right-hand plot shows the value of the objective function, the negative log-likelihood in the Bayesian inference problem.

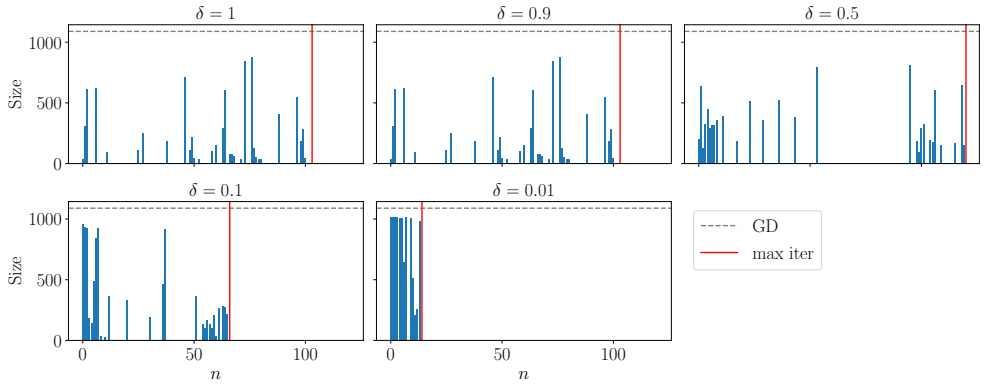


Fig. 5.3: Size of the matrix whose Cholesky factorisation must be computed for each iteration of gradient descent in the groundwater flow example from subsection 5.2, as the parameter δ is varied. The figure is for $N = 2$, $\dim(P) = 4$, but results for other parameter dimensions are similar. The dashed gray line shows the size of the (sparse) problem that must be solved for the classical approach, while the red line indicates the iteration number at which convergence was achieved.

on two challenging applied problems with favourable results compared to classical approaches. The chief advantages of the approach are that (i) gradients can be calculated at a lower cost than in classical approaches, (ii) that a global model for the gradient across parameter space is constructed, allowing for re-use of computational effort from previous iterations of gradient descent and (iii) that a full probability model is output, providing an error indicator that we used both to determine when to refine the approximation and to perform line searches.

Several possible avenues for future work present themselves. The first would be continuing to develop applications of this algorithm within optimisation, either by developing versions of more sophisticated gradient-based optimisation algorithms which exploit probabilistic gradients, or by extending the framework to obtain higher order information to accelerate the optimisation. Another would be to explore the use of probabilistic gradients in other applications. In particular, we note that while computing the MAP point is an important problem in Bayesian inference, sophisticated Markov-chain Monte-Carlo algorithms for sampling the posterior also make use of this information, and the posterior distribution over the gradient presented herein could straightforwardly be incorporated into such algorithms.

REFERENCES

[1] M. A. Álvarez, L. Rosasco, and N. D. Lawrence, *Kernels for vector-valued functions: a review*, Foundations and Trends in Machine Learning, 4 (2012), pp. 195–266, https://doi.org/10.1561/2200000036.

[2] L. Arriola and J. M. Hyman, *Sensitivity analysis for uncertainty quantification in mathematical models*, in Mathematical and statistical estimation approaches in epidemiology, Springer, 2009, pp. 195–247.

[3] A. V. Beddows, N. Kitwiroon, M. L. Williams, and S. D. Beevers, *Emulation and sensitivity analysis of the community multiscale air quality model for a UK ozone pollution episode*, Environmental Science & Technology, 51 (2017), pp. 6229–6236, https://doi.org/10.1021/acs.est.6b05873.

[4] P. Benner, S. Gugercin, and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM review, 57 (2015), pp. 483–531.
[5] P. Benner, E. Sachs, and S. Volkwein, *Model order reduction for PDE constrained optimization*, in Trends in PDE constrained optimization, Springer, 2014, pp. 303–326.
[6] A. Berlinet and C. Thomas-Agnan, *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, Springer US, 2004, https://doi.org/10.1007/978-1-4419-9096-9.
[7] L. T. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, *Real-time PDE-constrained Optimization*, SIAM, 2007.
[8] V. I. Bogachev, *Gaussian Measures*, vol. 62, American Mathematical Society Providence, 1998.
[9] J. F. Bonnans and A. Shapiro, *Perturbation analysis of optimization problems*, Springer Science & Business Media, 2013.
[10] K. Cheng, Z. Lu, C. Ling, and S. Zhou, *Surrogate-assisted global sensitivity analysis: an overview*, Structural and Multidisciplinary Optimization, 61 (2020), pp. 1187–1213, https://doi.org/10.1007/s00158-019-02413-5.
[11] I. Cialenco, G. E. Fasshauer, and Q. Ye, *Approximation of stochastic partial differential equations by a kernel-based collocation method*, Int. J. Comput. Math., 89 (2012), pp. 2543–2561, https://doi.org/10.1080/00207160.2012.688111.
[12] E. Cleary, A. Garbuno-Inigo, S. Lan, T. Schneider, and A. M. Stuart, *Calibrate, Emulate, Sample*, Journal of Computational Physics, (2020), p. 109716, https://doi.org/https://doi.org/10.1016/j.jcp.2020.109716.
[13] J. Cockayne, *Bayesian Probabilistic Numerical Methods*, PhD thesis, University of Warwick, 2019.
[14] J. Cockayne, C. J. Oates, T. J. Sullivan, and M. Girolami, *Bayesian probabilistic numerical methods*, SIAM Review, 61 (2019), pp. 756–789, https://doi.org/10.1137/17m1139357.
[15] H. B. Curry, *The method of steepest descent for non-linear minimization problems*, Q APPL MATH, 2 (1944), pp. 258–261, https://doi.org/10.1090/qam/10667.
[16] M. Drohmann and K. Carlberg, *The ROMES method for statistical modeling of reduced-order-model error*, SIAM/ASA Journal on Uncertainty Quantification, 3 (2015), pp. 116–145.
[17] L. Evans, *Partial Differential Equations*, American Mathematical Society, Mar. 2010, https://doi.org/10.1090/gsm/019.
[18] M. Fisher, J. Nocedal, Y. Trémolet, and S. J. Wright, *Data assimilation in weather forecasting: a case study in PDE-constrained optimization*, Optimization and Engineering, 10 (2009), pp. 409–426.
[19] R. FitzHugh, *Impulses and physiological states in theoretical models of nerve membrane*, Biophysical Journal, 1 (1961), pp. 445–466, https://doi.org/10.1016/s0006-3495(61)86902-6.
[20] C. Geyer, *Introduction to markov chain monte carlo*, Handbook of markov chain monte carlo, 20116022 (2011), p. 45.
[21] S. Girard, V. Mallet, I. Korsakissok, and A. Mathieu, *Emulation and Sobol′ sensitivity analysis of an atmospheric dispersion model applied to the Fukushima nuclear accident*, Journal of Geophysical Research: Atmospheres, 121 (2016), pp. 3484–3496, https://doi.org/10.1002/2015jd023993.
[22] M. Gunzburger, *Perspective in flow control and optimization (2003)*, SIAM, Philadelphia, 2002.
[23] E. Hairer, S. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer, 1993.
[24] D. Hartman and L. K. Mestha, *A deep learning framework for model reduction of dynamical systems*, in 2017 IEEE Conference on Control Technology and Applications (CCTA), IEEE, 2017, pp. 1917–1922.
[25] P. Hennig, M. A. Osborne, and M. Girolami, *Probabilistic numerics and uncertainty in computations*, J. R. Stat. Soc. A Stat., 471 (2015), pp. 20150142, 17, https://doi.org/10.1098/rspa.2015.0142.
[26] R. Herzog and K. Kunisch, *Algorithms for PDE-constrained optimization*, GAMM-Mitteilungen, 33 (2010), pp. 163–176, https://doi.org/10.1002/gamm.201010013.
[27] D. Higdon, M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne, *Combining field data and computer simulations for calibration and prediction*, SIAM Journal on Scientific Computing, 26 (2004), pp. 448–466.
[28] K. Ito and K. Kunisch, *Lagrange multiplier approach to variational problems and applications*, SIAM, 2008.
[29] R. Jin, W. Chen, and A. Sudjianto, *Analytical metamodel-based global sensitivity analysis and uncertainty propagation for robust design*, in SAE Technical Paper Series, SAE

International, Mar. 2004, https://doi.org/10.4271/2004-01-0429.
[30] M. C. Kennedy and A. O'Hagan, *Bayesian calibration of computer models*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63 (2001), pp. 425–464, https://doi.org/10.1111/1467-9868.00294.
[31] S. Lan, T. Bui-Thanh, M. Christie, and M. Girolami, *Emulation of higher-order tensors in manifold Monte Carlo methods for Bayesian inverse problems*, Journal of Computational Physics, 308 (2016), pp. 81–101.
[32] M. Mahsereci and P. Hennig, *Probabilistic line searches for stochastic optimization*, in Advances in Neural Information Processing Systems 28, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds., Curran Associates, Inc., 2015, pp. 181–189, http://papers.nips.cc/paper/5753-probabilistic-line-searches-for-stochastic-optimization.pdf.
[33] J. C. Newman III, A. C. Taylor III, R. W. Barnwell, P. A. Newman, and G. J.-W. Hou, *Overview of sensitivity analysis and shape optimization for complex aerodynamic configurations*, Journal of Aircraft, 36 (1999), pp. 87–96.
[34] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer New York, 2006, https://doi.org/10.1007/978-0-387-40065-5, https://doi.org/10.1007/978-0-387-40065-5.
[35] J. Oakley and A. O'Hagan, *Bayesian inference for the uncertainty distribution of computer model outputs*, Biometrika, 89 (2002), pp. 769–784.
[36] J. E. Oakley and A. O'Hagan, *Probabilistic sensitivity analysis of complex models: a Bayesian approach*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 66 (2004), pp. 751–769, https://doi.org/10.1111/j.1467-9868.2004.05304.x.
[37] C. J. Oates and T. J. Sullivan, *A modern retrospective on probabilistic numerics*, Statistics and Computing, 29 (2019), pp. 1335–1351, https://doi.org/10.1007/s11222-019-09902-z.
[38] M. Osborne, *Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature*, PhD thesis, PhD thesis, University of Oxford, 2010.
[39] H. Owhadi, *Bayesian numerical homogenization*, Multiscale Modeling & Simulation, 13 (2015), pp. 812–828, https://doi.org/10.1137/140974596.
[40] H. Owhadi, *Multigrid with rough coefficients and multiresolution operator decomposition from hierarchical information games*, SIAM Review, 59 (2017), pp. 99–149, https://doi.org/10.1137/15m1013894.
[41] H. Owhadi and L. Zhang, *Gamblets for opening the complexity-bottleneck of implicit schemes for hyperbolic and parabolic ODEs/PDEs with rough coefficients*, Journal of Computational Physics, 347 (2017), pp. 99–128, https://doi.org/10.1016/j.jcp.2017.06.037.
[42] R. Pulch, E. J. W. ter Maten, and F. Augustin, *Sensitivity analysis and model order reduction for random linear dynamical systems*, Mathematics and Computers in Simulation, 111 (2015), pp. 80–95.
[43] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, 2005, https://doi.org/10.7551/mitpress/3206.001.0001.
[44] M. Renardy, T.-M. Yi, D. Xiu, and C.-S. Chou, *Parameter uncertainty quantification using surrogate models applied to a spatial model of yeast mating polarization*, PLOS Computational Biology, 14 (2018), p. e1006181, https://doi.org/10.1371/journal.pcbi.1006181.
[45] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, *Design and analysis of computer experiments*, Statistical science, (1989), pp. 409–423.
[46] O. San and R. Maulik, *Extreme learning machine for reduced order modeling of turbulent geophysical flows*, Physical Review E, 97 (2018), p. 042322.
[47] O. San and R. Maulik, *Neural network closures for nonlinear model order reduction*, Advances in Computational Mathematics, 44 (2018), pp. 1717–1750.
[48] B. Sengupta, K. J. Friston, and W. D. Penny, *Efficient gradient computation for dynamical models*, NeuroImage, 98 (2014), pp. 521–527.
[49] M. Shafto, M. Conroy, R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne, and L. Wang, *Modeling, simulation, information technology & processing roadmap*, National Aeronautics and Space Administration, (2012).
[50] S. Sheriffdeen, J. C. Ragusa, J. E. Morel, M. L. Adams, and T. Bui-Thanh, *Accelerating PDE-constrained inverse solutions with Deep Learning and Reduced Order Models*, arXiv preprint arXiv:1912.08864, (2019).
[51] I. Sobol′, *Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates*, Mathematics and Computers in Simulation, 55 (2001), pp. 271–280, https://doi.org/10.1016/s0378-4754(00)00270-6.
[52] A. Stuart and A. Teckentrup, *Posterior consistency for Gaussian process approximations of Bayesian posterior distributions*, Mathematics of Computation, 87 (2018), pp. 721–753.
[53] A. M. Stuart, *Inverse problems: A Bayesian perspective*, Acta Numerica, 19 (2010), pp. 451–559, https://doi.org/10.1017/s0962492910000061.

[54] C. Villani, *Optimal Transport*, Springer Berlin Heidelberg, 2009, https://doi.org/10.1007/978-3-540-71050-9.
[55] H. Wendland, *Scattered Data Approximation*, Cambridge University Press, Dec. 2004, https://doi.org/10.1017/cbo9780511617539.