Statistics and Computing (2019) 29:1285–1295
https://doi.org/10.1007/s11222-019-09899-5

Adaptive step-size selection for state-space probabilistic differential equation solvers

Oksana A. Chkrebtii[1] · David A. Campbell[2]

Published online: 28 September 2019

Abstract
When models are defined implicitly by systems of differential equations with no closed-form solution, small local errors in finite-dimensional solution approximations can propagate into deviations from the true underlying model trajectory. Some recent perspectives in quantifying this uncertainty are based on Bayesian probability modeling: a prior is defined over the unknown solution and updated by conditioning on interrogations of the forward model. Improvement in accuracy via grid refinement must be considered in order for such Bayesian numerical methods to compete with state-of-the-art numerical techniques. We review the principles of Bayesian statistical design and apply these to develop an adaptive probabilistic method to sequentially select time-steps for state-space probabilistic ODE solvers. We investigate the behavior of local error under the adaptive scheme which underlies numerical variable step-size methods. Numerical experiments are used to illustrate the performance of such an adaptive scheme, showing improved accuracy over uniform designs in terms of local error.

Keywords Uncertainty quantification · Differential equations · Statistical design · Numerical methods · Data assimilation

1 Introduction

The fact that forward models based on the numerical solution of differential equations are subject to error has been studied for a long time. One of the ways in which the field of numerical analysis quantifies this error is by approximating global error bounds on the solution at each spatio-temporal location (e.g., Butcher 2008). The size of the error bound can be controlled by the choice of numerical method and by changing the number and location of discretization grid points. An important advantage of this approach is that it provides an upper limit on the deviation of the approximation from the true state. In other words, it can be shown that the solution lies *somewhere* within this possibly large error bound. A drawback of this approach is the lack of information about the *relative likelihood* of the solution being in a given region within the error bound. It is therefore not clear how to account for this error within statistical inference (the inverse problem) given observed data. The developing field of probabilistic numerics, which ultimately seeks a Bayesian view of problems in numerical analysis, calls for modeling the uncertainty in the forward problem by describing, either exactly or approximately, the relative likelihood of the solution across the phase space.

At this stage, it is helpful to define some notation. We primarily deal with the solution of the ordinary differential equation (ODE) initial value problem,

$$\begin{cases} Du = f\left(t, u\right), & t \in (0, L], \\ u \quad = u_1, & t = 0, \end{cases} \tag{1}$$

where u_1 is a vector of initial states, D is a linear differential operator and $f : [0, L] \times \mathbb{R}^p \rightarrow \mathbb{R}^p$ is a vector field that is Lipschitz continuous in the second argument. The exact solution at time t is denoted by $u^\dagger(t; u_1)$. The notation $u(t; u_1)$ is reserved for describing a probability model for the uncertainty in the unknown solution to (1). A single sample from this process is denoted by a tilde, e.g., $\tilde{u}(t; u_1)$. In order to relate our results to the numerical analysis literature, we denote numerical solver output by $\hat{u}(t; u_1)$. In all these

Handling Editor: T. J. Sullivan.

✉ Oksana A. Chkrebtii
oksana@stat.osu.edu

David A. Campbell
davecampbell@math.carleton.ca

[1] Department of Statistics, The Ohio State University, 1958 Neil Avenue, Columbus, OH 43210, USA

[2] School of Mathematics and Statistics, Carleton University, 1125 Colonel By Drive, Ottawa, ON K1S 5B6, Canada

cases, the dependence on the initial condition u_1 will be omitted when it is clear from the context. This paper describes a probabilistic discretization technique, performed over a partition $s_1, \dots, s_N \in [0, L]$ of the domain of integration. Objects evaluated at these time points are either denoted explicitly as, e.g., $u(s_i)$, or by using subscript notation, e.g., $u_i := u(s_i)$, where the subscript i indicates that u is evaluated at time s_i. When a quantity is evaluated over multiple subsequent time points, or a range of partitions of the domain is desired, we use a semicolon in the subscript between the first and last indices, e.g., $u_{i_1:i_2}$ and $s_{i_1:i_2}$, respectively. For processes or quantities that are updated sequentially, such as the probability model on the states or the mean and covariance, the update number is denoted by a superscript, e.g., u^i, m^i, C^i, respectively. Finally, when multiple parallel updates of these quantities are required, additional superscripts in brackets indicate the number of the ensemble member, e.g., $u^{i(j)}$, $m^{i(j)}$.

The use of probability models for uncertainty quantification in numerical problems was first proposed by Diaconis (1988), O'Hagan (1992) and Skilling (1991). The problem consists of solving the definite integral $u(t) = \int_0^t g(x)dx, t \in [0, L]$, where $g : [0, L] \to \mathbb{R}^p$ is a fully specified integrable function on $[0, L]$. First, a prior model is defined jointly on the unknown function u and its derivative g, which describes the analyst's belief before any analysis is conducted and incorporates any definitive knowledge in the form of constraints. Exact time-derivatives of u are obtained over a grid $0 = s_1 < s_2 < \cdots < s_N = L$ by evaluating the function g at those points. The posterior over u given the derivatives $g(s_1), \dots, g(s_i)$ describes the uncertainty in the solution conditional on the information about g. Because this is a function estimation problem with noise-free observations, the convergence properties of the posterior to the Dirac measure centered at u are available (e.g., Stuart 2010) as the number of evaluation points grows within the domain $[0, L]$.

Differential equation models describe system states implicitly in terms of rates of change with respect to spatial and temporal variables. Solving the ODE initial value problem (1), although related to integration, is made more complex by the implicit dependence of the state on its own derivatives. Indeed, for any $t_1 < t_2$, the solution $u(t_2)$ is a function of $u(t_1)$ that does not depend on $u(\tau), \tau \in [0, t_1)$ (a deterministic Markov property, e.g., Jazwinski 1970). The modern ideas for Bayesian numerical analysis of differential equations were established in an innovative paper by Skilling (1991). Interest lies in inferring the unknown deterministic function $u^\dagger : [0, L] \to \mathbb{R}^p$, defined implicitly via the ODE initial value problem (1). Skilling's contribution was the idea of modeling uncertainty in the fixed but unknown solution using the Bayesian paradigm, by updating prior belief about the solution by conditioning on evaluations of the forward model. A prior process is defined jointly on u and Du, and marginal state samples $\tilde{u}^0(s_1), \dots, \tilde{u}^0(s_N)$ are generated (not necessarily in order) over a grid $s_1, \dots, s_N \in [0, L]$. Evaluating the vector field f at these sampled values yields a sequence which can be modeled as noisy evaluations of Du and hence used to update our prior belief about $u^\dagger$. Crucially, however, this model specification does not directly take into account the Markovian structure of the ODE solution (Chkrebtii et al. 2016b), so that convergence to the true solution is guaranteed only when the model is explicit (i.e., equivalent to a direct integration problem). A reformulation of this procedure based on sequential updating of the prior (Hennig and Hauberg 2014; Schober et al. 2014) resolves the problem, but lacks the flexibility to capture the non-Gaussian structure of the error. This is especially problematic when system solutions are restricted to low-dimensional manifolds.

The formalism of Cockayne et al. (2017) provides an elegant alternative to the above methods via a collocation-based approach, where the posterior over the solution trajectory is constrained to satisfy $Du(s_i) = f(s_i, u(s_i))$ at collocation points $s_1, \dots, s_N \in [0, L]$. A compelling argument for this approach is that *exact* knowledge about the true solution at the collocation points is enforced. A disadvantage is that direct sampling from this probability measure is infeasible except when the ODE can be reformulated as an explicit integration problem (Wang et al. 2018). Furthermore, Wang et al. (2018) demonstrate that an exact likelihood cannot be obtained in general due to the implicit specification of ODE initial value problems.

The view we take in this paper, proposed in Chkrebtii et al. (2016a), is a state-space Bayesian numerical method. While it requires likelihood modeling at each discretization grid point, this model can be set up in a way that reflects our belief about the solution at each point. This approach admits uncertainty estimates that are flexible (non-Gaussian) at a computational cost, based on the number of operations required, that is proportional to that of a similarly sampled numerical solver (Chkrebtii et al. 2016a). We are ultimately interested in the posterior over the solution conditional on the specification of the initial value problem (1), determined by the vector field f and the initial condition u_1. Since we cannot condition directly on the model specification, auxiliary variables $a_{1:N}$ are used to interrogate the model at times $s_{1:N}$ where $0 = s_1 < \cdots < s_N = L$. Each auxiliary variable $a_i = \left(u_i^{i-1}, f(s_i, u_i^{i-1})\right), i = 1, \dots, N$ consists of two components: the posterior predictive process at time s_i over u obtained after $i - 1$ algorithm iterations denoted u_i^{i-1}, and the corresponding vector field $f_i = f(s_i, u_i^{i-1})$. The posterior density over the state u evaluated at any desired time points in $[0, L]$ can be written as,

$$\pi(u \mid f(\cdot), u_1) \\ = \int \pi(u, a_{1:N} \mid f(\cdot), u_1)\, \mathrm{d}a_{1:N}$$

$$\begin{aligned}
&\propto \int p\left(f_1 \mid u_1\right) \\
&\quad p\left(u_2 \mid f_1, u_1\right) p\left(f_2 \mid u_2\right) \\
&\quad p\left(u_3 \mid f_{1:2}, u_1\right) p\left(f_3 \mid u_3\right) \\
&\quad \cdots\ p\left(u_N \mid f_{1:N-1}, u_1\right) p\left(f_N \mid u_N\right) \\
&\quad p\left(u \mid f_{1:N}, u_1\right) \mathrm{d}a_{1:N}, \\
&= \int p\left(u \mid f_{1:N}, u_1\right) \\
&\quad \prod\nolimits_{i=1}^{N} \left\{p\left(f_i \mid u_i\right) p\left(u_i \mid f_{1:i-1}, u_1\right)\right\} \mathrm{d}a_{1:N}. \qquad (2)
\end{aligned}$$

The integrand is formed by an iterative application of analysis steps $p\left(f_i \mid u_i\right)$ and forecast steps $p\left(u_i \mid f_{1:i-1}, u_1\right)$, followed by a final smoothing step $p\left(u \mid f_{1:N}, u_1\right)$ (see, e.g., Wikle and Berliner 2007 for an overview of this approach in the filtering literature). Assuming a Gaussian prior and likelihood, the densities $p\left(u_i \mid f_{1:i-1}, u_1\right)$ are Gaussian, while $p\left(f_i \mid u_i\right)$ are generally not available in closed-form for $i > 1$. In other words, (2) is a continuous mixture of Gaussian forecast densities weighted by the corresponding analysis densities. Section 2.1 provides additional details on this approach, and Algorithm 1 summarizes a sequential Monte Carlo procedure developed in Chkrebtii et al. (2016a) for sampling from (2).

Numerical algorithms for solving ODE initial value problems can be designed to adapt step lengths in order to use computational resources efficiently by controlling a chosen criterion, such as local truncation error. As a result, some type of step-size control is employed in most numerical solver software (Shampine 2005). We show that applying principles of statistical design for state-space probabilistic solvers can yield analogously effective and interpretable criteria for sequential step length selection.

The paper is organized as follows. Section 2 describes the state-space Bayesian numerical solver (hereafter called UQDE) in detail, discusses approaches to adaptive step length selection for one-step numerical solvers, and reviews the principles of Bayesian experimental design. Our new approach and main results are provided in Sect. 3, examples and numerical results are provided in Sect. 4, and conclusions are discussed in Sect. 5. MATLAB implementation of the proposed approach is available at github.com/ochkrebtii/uqdes-enkf-design along with code to reproduce all figures.

2 Background

This section details the state-space uncertainty quantification approach developed in Chkrebtii et al. (2016a) for the ODE initial value problem (1). We then provide a general overview of the goals and strategies of adaptive step-size selection methods employed in the numerical analysis literature for one-step solvers. The last subsection reviews the topic of statistical design of experiments from a Bayesian perspective.

2.1 State-space probabilistic solver

The probabilistic numerical method developed in Chkrebtii et al. (2016a) for the forward model (1) can be viewed as a continuous-time filter (e.g., Jazwinski 1970). This formulation extends readily to PDEs where explicit integration tools can be used over the spatial component of the domain. A large class of numerical approaches to solving ODEs are based on linearization of the implicitly-defined forward model. The probabilistic analog is the following assumed error model,

$$\begin{aligned}
f_i &:= f(s_i, u^{i-1}(s_i)) \\
&= Du^{\dagger}(s_i) + \xi(s_i),\ i = 1, \ldots, N \qquad (3)
\end{aligned}$$

where the error term ξ is the source of model uncertainty. Throughout this work, we will assume that $\xi(s_i)$ are independent, zero-mean Gaussian random variables with variance $Q(s_i, s_i)$. In the context of the unknown solution of a deterministic ODE initial value problem, the stochastic model (3) is a description of our incomplete knowledge about the explicit solution $u^{\dagger}$ from a Bayesian perspective.

The first step is to define a joint prior model on the unknowns. Specifically, we assume a Gaussian process (GP) prior over $(Du, u)^{\top}$ with mean function $\left(Dm^0, m^0\right)^{\top}$ and covariance operator that has block components $C_{1,1} = DC^0D^*$, $C_{2,2} = C^0$, and $C_{1,2} = C^*_{2,1} = DC^0$. Here, C^0 is a prior covariance operator and the asterisk represents the adjoint of an operator. The GP model covariance is selected to reflect our knowledge about the function space where the solution lies. Sequentially updating the prior over $(Du, u)^{\top}$ by conditioning on $f_{1:i}$ yields the updated distribution,

$$\begin{aligned}
&\left(\begin{matrix} Du(t_k) \\ u(t_\ell) \end{matrix} \,\middle|\, f_{1:i}\right) \\
&\quad \sim \mathcal{GP}\left\{\left(\begin{matrix} Dm^i(t_k) \\ m^i(t_\ell) \end{matrix}\right), \left(\begin{matrix} DC^i(t_k, t_k)D^* & DC^i(t_k, t_\ell) \\ C^i(t_\ell, t_k)D^* & C^i(t_\ell, t_\ell) \end{matrix}\right)\right\}, \qquad (4)
\end{aligned}$$

where means and covariances can be defined recursively as,

$$m^i(t) = m^{i-1}(t) + K^i(t, s_i)\left\{f_i - Dm^{i-1}(s_i)\right\} \qquad (5)$$

$$\begin{aligned}
C^i(t_k, t_\ell) &= C^{i-1}(t_k, t_\ell) \\
&\quad - K^i(t_k, s_i)DC^{i-1}(s_i, t_\ell) \qquad (6) \\
K^i(t, s_i) &= C^{i-1}(t, s_i)D^*\Big(Q(s_i, s_i) \\
&\quad + DC^{i-1}(s_i, s_i)D^*\Big)^{-1}, \qquad (7)
\end{aligned}$$

and $Q(s_i, s_i)$ denotes the variance of the discrepancy between f and Du and at time s_i. For our numerical experiments, we choose $Q(s_i, s_i)$ to be the predictive variance, $DC^{i-1}(s_i, s_i)D^*$, over Du at time s_i, although this modeling choice can be application-dependent.

The sequential updating procedure is summarized in Algorithm 1. The algorithm yields J Monte Carlo samples from the posterior (2) at T desired evaluation time points $x_{1:T} \in [0, L]^T$. Although each Monte Carlo sample $j = 1, \ldots, J$ can be computed in parallel, computational savings can also be obtained by pooling covariance updates when the grid spacing does not change between draws. In order to exploit these computational savings and to ensure consistent interpretation of uncertainty between draws, in what follows, our proposed approach will assign each Monte Carlo sample a common, but adaptively-selected design.

Algorithm 1 Sequential Monte Carlo algorithm to sample from the posterior (2) at evaluation points $x_{1:T} \in [0, L]^T$ and ordered discretization grid locations $s_{1:N} \in [0, L]^N$

```
Define $\mathbf{t} = (s_{1:N}, x_{1:T})$ where $0 = s_1 < s_2 < \cdots < s_N = L$. The functions $C^0, m^0$, and $Q$ are defined in Sect. 2.1. The state $u$, its initial condition $u_1$, vector field $f$, and the linear differential operator $D$, are defined as in Eq. (1).
Compute $f_1^{(j)} = f(s_1, u_1)$, $j = 1, \ldots, J$
for $j = 1 : J$ do
    for $i = 1 : N$ do
        Given $f_i^{(j)}$, compute the following quantities recursively,
            $K^i(\mathbf{t}, s_i) = C^{i-1}(\mathbf{t}, s_i)D^* \left\{Q(s_i, s_i) + DC^{i-1}(s_i, s_i)D^*\right\}^{-1}$,   (8)
            $m^{i(j)}(\mathbf{t}) = m^{i-1(j)}(\mathbf{t}) + K^i(\mathbf{t}, s_i)\left\{f_i^{(j)} - Dm^{i-1(j)}(s_i)\right\}$,   (9)
            $C^i(\mathbf{t}, \mathbf{t}) = C^{i-1}(\mathbf{t}, \mathbf{t}) - K^i(\mathbf{t}, s_i)DC^{i-1}(s_i, \mathbf{t})$;   (10)
        if $i < N$ then
            Sample $\tilde{u}_{i+1}^{i(j)}$ from
                $\mathcal{GP}\left(m^{i(j)}(s_{i+1}), C^i(s_{i+1}, s_{i+1})\right)$
            Compute $f_{i+1}^{(j)} = f\left(s_{i+1}, \tilde{u}_{i+1}^{i(j)}\right)$
        end if
    end for
end for
For each $j = 1, \ldots, J$, sample and retain a draw $\tilde{u}_{1:T}^{N(j)}$ from $\mathcal{GP}\left(m^{N(j)}(x_{1:T}), C^N(x_{1:T}, x_{1:T})\right)$ to approximate any desired posterior functionals.
```

To illustrate the influence of grid spacing on probabilistic solver uncertainty, consider the chaotic Kuramoto-Sivashinsky model of a reaction-diffusion system (Kuramoto and Tsuzuki 1976),

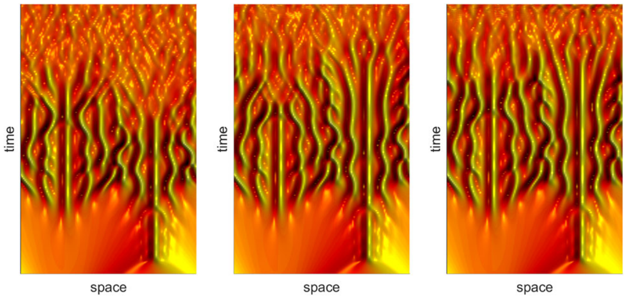

Fig. 1 Point-wise sample mean of 10 realizations from the posterior trajectory (2) of the spatially-discretized Kuramoto-Sivashinsky PDE initial value problem (11) on the domain $x \in [0, 32\pi]$, $t \in [0, 150]$. The number of time-steps, from left to right, is 1000, 2000, and 3000, respectively. Sensitivity to perturbations introduces discretization uncertainty that manifests as variability between sample paths over time (blurry regions). Trajectories are initially very similar (well-defined regions) until they eventually diverge

$$\begin{cases} \frac{\partial}{\partial t}u = -u\frac{\partial}{\partial x}u - \frac{\partial^2}{\partial x^2}u - \frac{\partial^4}{\partial x^4}u, \\ \qquad x \in [0, 32\pi], \ t \in (0, 150] \\ u \ = \cos\left(\frac{x}{16}\right)\left\{1 + \sin\left(\frac{x}{16}\right)\right\}, \\ \qquad x \in [0, 32\pi], \ t = 0. \end{cases} \tag{11}$$

The model was discretized along the spatial domain (using an equally spaced grid of size $N_s = 128$) and projected into the Fourier domain. Defining $\bar{u} = F(u)$ to be the discrete Fourier transform of u evaluated at an even number N_s of spatial locations, this becomes,

$$\begin{cases} \frac{\partial}{\partial t}\bar{u} = -\frac{ik}{2}\bar{u}^2 + \left(k^2 - k^4\right)\bar{u}, \ t \in (0, 150] \\ \qquad k \in \{0, \ldots, N_s/2 - 1, 0, -N_s/2 + 1, \ldots, -1\}/16, \\ \bar{u} \ = F\left(\cos\left(\frac{x_i}{16}\right)\left\{1 + \sin\left(\frac{x_i}{16}\right)\right\}\right), t = 0, \\ \qquad i = 1, \ldots, N_s. \end{cases} \tag{12}$$

Following Kassam and Trefethen (2005), the integrating factor method is applied in order to retain only the purely nonlinear terms of the 128-dimensional ODE initial value problem (12). Complex dynamics emerge in this model as energy is transferred from low to high wave numbers, which manifests itself through the visible onset of chaotic dynamics, as shown in Fig. 1. The figure shows an average of $J = 10$ trajectory images drawn from the posterior over the state obtained via Algorithm 1. Clearly defined patterns near the base of each figure indicate trajectories that are almost exactly the same, while cloudy patterns near the top of each figure indicate differences between realizations. Greater solver accuracy due to finer discretization grid spacing (from left to right) pushes the onset of chaos further forward in time. The motivation of the present work is to improve accuracy by arranging a fixed number of discretization grid points to maximize the amount of information obtained about the solution.

2.2 Adaptive step-size selection for numerical methods

In numerical analysis, step-size control is the problem of choosing a grid spacing that reduces the local truncation error. Local error is defined as the error incurred at a given step of a numerical solver under the assumption that the step begins at the exact solution and that there is no round-off error. In practice, this discrepancy is computed as the difference between the output of the numerical solver $\hat{u}_i$ at time s_i and the exact solution obtained by integrating the ODE from time step s_{i-1} with initial condition given by $\hat{u}_{i-1}$, the output of the numerical solver at time s_{i-1}. This computable estimate, le_i, of the local error is given by:

$$le_i = \hat{u}_i - u^{\dagger}(s_i; \hat{u}_{i-1}), \tag{13}$$

where $\hat{u}_i$ is the numerical solution at step i and $u^{\dagger}(s_i; \hat{u}_{i-1})$ is the exact solution at s_i of the forward model (1) on the domain $[s_{i-1}, s_i]$ when the initial condition at s_{i-1} is $\hat{u}_{i-1}$. The local error typically has the following solver-dependent form:

$$le_i = h_i^{p+1}\phi\left(s_i, \hat{u}_i\right) + O\left(h_i^{p+2}\right), \tag{14}$$

where $h_i = s_i - s_{i-1}$ denotes the step length and the principal error function ϕ depends both on the numerical method and the functional form of the vector field. Based on this approximation, a variety of strategies can be adopted, from placing an upper bound on the local error at each step and halving the step size until this criterion is met, to choosing a step length out of several candidates with the smallest local error. An overview of these techniques for one-step methods is provided in Shampine (2005).

In contrast, controlling the global error, the difference between the approximation and the exact solution, requires numerically solving the system of differential equations more than once, which is often impractical and will not be considered in the present work. We note, however, that although the rate of error propagation depends on the form of the forward model, for most stable (non-stiff) ODE initial value problems, ensuring that local error remains very low at each step (e.g., under a small maximum step size) will translate to smaller global error overall, as will be illustrated in subsequent numerical experiments.

2.3 Bayesian maximum entropy designs

Bayesian statistical design is the problem of choosing an input s from a set $\mathcal{S}$ at which to measure data y, which belongs to a sample space $\mathcal{Y}$ and depends on an unknown quantity $u \in \mathcal{U}$. In the language of decision theory (see, for example, Bernardo and Smith 1994 p. 148, Chaloner and Verdinelli 1995; Sebastiani and Wynn 2000), the choice of s is called a decision and is associated with a user-defined utility function $v : \mathcal{U} \times \mathcal{Y} \times \mathcal{S} \rightarrow \mathbb{R}$ that reflects the gain associated with the decision. Choice of the utility function v depends on the goal of the inference problem and the application. An overview of several popular utility functions is provided in Chaloner and Verdinelli (1995).

Because u is unknown and y has not yet been observed, the objective function is the expectation of the utility with respect to their joint density. A design point s is selected if it maximizes the expected utility, v, of additional information $y(s)$ about the unknown quantity u,

$$s = \underset{s \in \mathcal{S}}{\arg\max} \int_{\mathcal{Y}} \mathrm{E}_{u|y,s}\, v\left(u, y(s), s\right) \mathrm{d}y. \tag{15}$$

Here, we consider a utility function motivated by information theory. Define the negative entropy $\mathrm{Ent}(w) := \mathrm{E}_w\left\{-\log p_w(w)\right\}$, of a random variable w with density p_w, and the utility function $v(u, y(s), s) = \mathrm{Ent}(u) - \mathrm{Ent}\left(u \mid y(s)\right)$. The resulting optimization problem is,

$$s = \underset{s \in \mathcal{S}}{\arg\max} \int_{\mathcal{Y}} \left\{\mathrm{Ent}(u) - \mathrm{Ent}\left(u \mid y(s)\right)\right\} \mathrm{d}y. \tag{16}$$

The integrand in the above expression is known as the *Kullback–Leibler (KL) entropy* (Kullback and Leibler 1951), a non-symmetric discrepancy between the prior μ^{pr} and posterior μ^{pt} measures over u, denoted by $\mathcal{D}\left(\mu^{pt} \mid\mid \mu^{pr}\right)$. The expression for the KL entropy between finite-dimensional prior and posterior Gaussian measures is well-known. When μ^{pr} and μ^{pt} are infinite-dimensional Gaussian processes with means m^{pr}, m^{pt} and covariances C^{pr}, C^{pt}, $\mathcal{D}\left(\mu^{pt} \mid\mid \mu^{pr}\right)$ has the form (e.g., Alexanderian et al. 2016),

$$\begin{aligned}\mathcal{D}\left(\mu^{pt} \mid\mid \mu^{pr}\right) = \frac{1}{2}\Big[&\log\det\left\{I + \left(C^{pr}\right)^{1/2} D^{*} D \left(C^{pr}\right)^{1/2}\right\} \\ &- \mathrm{tr}\left(D^{*} D C^{pt}\right) + \langle m^{pt} - m^{pr}, m^{pt} - m^{pr}\rangle_{\left(C^{pr}\right)^{-1}}\Big],\end{aligned} \tag{17}$$

where $\langle \cdot, \cdot \rangle_C$ represents the weighted inner product between two square-integrable functions on the domain $[0, L]$ and $||\cdot||_C$ represents its induced norm. In the following section, we discuss how this design principle can be applied to the choice of step length for Algorithm 1.

3 Methods

Our goal is to maximize the information gain about the unknown model state $u \in \mathcal{U}$ which solves the initial value problem (1), by choosing the placement of model interrogations sequentially. This is equivalent to choosing the length of the time step in Algorithm 1. The present application is well-suited to such a sequential approach because interrogating the model at candidate locations, that is, evaluating the vector field $f(\cdot)$ in Eq. (1), is computationally less expensive than making a single update of the probabilistic solver.
Though the criterion requires evaluating the vector field once for every parallel Monte Carlo sample generated, we recommend that if a large Monte Carlo sample is desired, a sub-sample of the J evaluations be used to approximate the expected utility function (16).

The proposed algorithm will begin by training the model over a small number S of initial steps of equal size h until it reaches a pre-determined starting point $a = Sh$ for the adaptation. In our numerical experiments, we chose this start time to be $a = 0.1L$, where L is the upper limit of integration in (1). For each iteration $i > S$, the algorithm will consider a small number P of candidate points on each interval $I_i = (s_i, s_i + h_{\max}]$, where $h_{\max}$ is a maximum allowed step length chosen by the user. We recommend taking $h_{\max}$ to be the step size associated with the lowest resolution one would be willing to tolerate, in terms of the local error. The candidate locations, denoted by $s_i^1, \ldots, s_i^P$, form an ordered partition of the domain I_i such that $s_i < s_i^1 < \cdots < s_i^P = s_i + h_{\max}$. In our numerical experiments, these are chosen to be equally spaced. Although a choice of $P = 2$ ensures the easiest visualization of the discretization grid for our numerical experiments, P can be as large as desired subject to computational considerations. A Monte Carlo estimate of the expected utility function (16) will be optimized over this partition to choose the next design point. Section 3.2 will provide additional recommendations and details of the settings used in our numerical experiments.

3.1 Objective function

Recall that the posterior (2) over u given $f(\cdot)$, u_1 is a continuous mixture of Gaussians, weighted by the analysis densities $p(f_i \mid u_i)$. Therefore, at each step, the integrand in (16) can, at least in principle, be computed exactly. The outer integral with respect to the marginal density $p(f_i \mid u_i)$ of the "data" in (16) can be approximated based on a Monte Carlo sample of trajectories of size J. Thus, the objective function in (16) becomes,

$$\int \mathcal{D}\left(\mu^{u|f_{1:i+1}} \mid\mid \mu^{u|f_{1:i}}\right) df_{i+1},$$

$$\approx \frac{1}{J}\sum\nolimits_{j=1}^{J} \mathcal{D}\left(\mu^{u|f_{1:i+1}^{(j)}} \mid\mid \mu^{u|f_{1:i}^{(j)}}\right), \tag{18}$$

where $\{f_i^{(j)}\},\ j = 1, \ldots, J,\ i = 1, \ldots, N$ is an ensemble of J vector field evaluations computed by running Algorithm 1 in parallel.

In principle, we wish to evaluate the expected utility function (18) at the P candidate points within the interval $I_i = (s_i, s_i + h_{\max}]$ for each iteration $i > S$. In practice, this task must be balanced with computational considerations. The Kullback–Leibler divergence in (18) consists of three terms of different importance in the context of step-size selection. The first two terms depend on the trajectory-wise covariance functions C^i and C^{i+1} computed before and after a single update. While these functions depend on the step size, they are independent of the state and vector field. Indeed, numerical experiments suggest that these terms change little with step size relative to the third term of the expression, which depends directly on the state. Since the covariances in the first two terms typically remain relatively stable within the interval I_i, we assume that the contribution of these terms toward distinguishing between the candidate points is negligible. The third term has the most direct dependence on the forward model and is assumed to be the main source of variability in the objective function over the interval I_i. This third term can be expressed as,

$$\langle m^i - m^{i-1}, m^i - m^{i-1} \rangle_{C^{i-1^{-1}}}$$

$$= \left|\left| K^{i-1}(\cdot, s) \left\{ f(s) - Dm^{i-1}(s) \right\} \right|\right|^2_{C^{i-1}(\cdot,\cdot)^{-1}},$$

$$= \left|\left| \left\{ Q(s, s) + DC^{i-1}(s, s)D^* \right\}^{-1} \right.\right.$$

$$\left.\left. \cdot \left\{ f(s) - Dm^{i-1}(s) \right\} \right|\right|^2_{\Lambda}, \tag{19}$$

with weight function $\Lambda(t, t)$ given by,

$$\Lambda(t, t) = DC^{i-1}(s, t)C^{i-1}(t, t)^{-1}C^{i-1}(t, s)D^*. \tag{20}$$

Expression (19) can be simplified to,

$$\int_a^L \Lambda(t, t)\mathrm{d}t \left\{ Q(s, s) + DC^{i-1}(s, s)D^* \right\}^{-2}$$

$$\cdot \left\{ f(s) - Dm^{i-1}(s) \right\}^2, \tag{21}$$

corresponding to the squared difference between the next model evaluation $f(s)$ and the predicted mean $Dm^{i-1}(s)$, scaled by the strength of correlation between $u(t)$ and $f(s)$, and inversely proportional to the error variance $Q(s, s)$ of Du at time s. This intuitively describes the idea of choosing a design point $s_{i+1} \in I_i$ in such a way that the information obtained at this point has not been well-explained by the updated model up to that point. This yields the following rule for selecting the next discretization grid location,

$$
\begin{aligned}
s_{i+1} = \arg\max_{s \in \{s_i^1, \ldots, s_i^P\}} \\
\int_a^L \Lambda(t, t)\mathrm{d}t \left(Q(s, s) + DC^{i-1}(s, s)D^* \right)^{-2} \\
\cdot \frac{1}{J} \sum_{j=1}^J \left(f^{(j)}(s) - Dm^{i-1(j)}(s) \right)^2 ,
\end{aligned}
\tag{22}
$$

which maximizes the information provided by the vector field at the new location, as quantified by approximation (18) to the expected utility function (16). Note that the weight $\int_a^L \Lambda(t, t)\mathrm{d}t$ cannot in general be computed in closed-form. Although using a numerical approximation of this quantity may appear undesirable in the context of a probabilistic numerical method, this quantity only serves to select the step length of the algorithm. In other words, discretization uncertainty associated with its use is still accounted for by the model in a separate calculation.

3.2 Implementation details

Implementing a state-space based probabilistic method requires choosing a suitable covariance structure and its hyperparameters. Chkrebtii et al. (2016a) suggest that the covariance structure should reflect our a-priori assumptions about the smoothness of the unknown solution $u^\dagger$. This, however, must be balanced with computational efficiency considerations because judicious choices can significantly speed up implementation. For example, when the covariance has bounded support, updating the mean and covariances (23) - (25) only requires updating of a subset of their elements from the previous iteration. Thus, our numerical experiments employ a uniform covariance structure with half-width (length-scale) equal to the maximum allowed step length $h_{\max}$. The covariance prior precision is chosen to be small (0.1) in order to reflect our initial uncertainty about the range of the solution. When observations on the states are available, these hyperparameters can also be learned from the data within the inverse problem (e.g., Chkrebtii et al. 2016a).

Algorithm 2 Sequential grid-adaptive Monte Carlo algorithm to sample from the posterior (2) at T evaluation points $x_{1:T} \in [0, L]^T$, ordered grid points $s_{1:S} \in [0, L]^S$, and candidate discretization grid points $s_{(S+1):N}^{1:P} \in [0, L]^{(N-S)P}$

```
Define $\mathbf{t} = \left( s_{1:S}, s_{(S+1):N}^{1:P}, x_{1:T} \right)$, where $0 = s_1 < s_2 < \cdots < s_N = L$, for $i > S$ the partitions are $s_i < s_i^1 < \cdots < s_i^P = s_i + h_{\max}$, and for $i \leq S$ we have $s_i = s_{i-1} + h$. The functions $C^0$, $m^0$, and $Q$ are defined in Sect. 2.1. The state $u$, vector field $f$, and the linear differential operator $D$, are defined in Eq. (1).
Compute $f_1^{(j)} = f(s_1, u_1), \; j = 1, \ldots, J$
for $i = 1 : N$ do
    for $j = 1 : J$ do
        Given $f_i^{(j)}$, compute the following quantities recursively,
            $K^i(\mathbf{t}, s_i) = C^{i-1}(\mathbf{t}, s_i)D^* \left\{ Q(s_i, s_i) + DC^{i-1}(s_i, s_i)D^* \right\}^{-1},$   (23)
            $m^{i(j)}(\mathbf{t}) = m^{i-1(j)}(\mathbf{t}) + K^i(\mathbf{t}, s_i) \left\{ f_i^{(j)} - Dm^{i-1(j)}(s_i) \right\},$   (24)
            $C^i(\mathbf{t}, \mathbf{t}) = C^{i-1}(\mathbf{t}, \mathbf{t}) - K^i(\mathbf{t}, s_i)DC^{i-1}(s_i, \mathbf{t});$   (25)
    end for
    if $i < N$ then
        if $i > S$ then
            Choose the next grid point by maximizing the objective function
                $s_{i+1} = \arg\max_{s \in \{s_i^1, \ldots, s_i^P\}} \int_a^L \Lambda(t, t)\mathrm{d}t \left\{ Q(s, s) + DC^{i-1}(s, s)D^* \right\}^{-2} \cdot \frac{1}{J} \sum_{j=1}^J \left( f^{(j)}(s) - Dm^{i-1(j)}(s) \right)^2$   (26)
        end if
        for $j = 1 : J$ do
            Sample $\tilde{u}_{i+1}^{i(j)}$ from
                $\mathcal{GP}\left( m^{i(j)}(s_{i+1}), C^i(s_{i+1}, s_{i+1}) \right)$
            Compute $f_{i+1}^{(j)} = f\left( s_{i+1}, \tilde{u}_{i+1}^{i(j)} \right)$
        end for
    end if
end for
For each $j = 1, \ldots, J$, sample and retain a draw $\tilde{u}_{1:T}^{N(j)}$ from $\mathcal{GP}\left( m^{N(j)}(x_{1:T}), C^N(x_{1:T}, x_{1:T}) \right)$ to approximate any desired posterior functionals.
```

3.3 Relationship between adaptation and local truncation error

An important consideration is how to quantify the reduction in the error associated with different discretizations of a state-space based probabilistic algorithm. This can be done by relating the step length adaptation based on Bayesian experimental design to numerical methods that control a computable estimate of local truncation error. For this, we generalize the concept of truncation error to stochastic models of uncertainty. We propose to consider the expected local error:

$$
ele_i = E_{u|f(\cdot),u_1} \left\{ u_i - u^\dagger(s_i; u_{i-1}) \right\} , \tag{27}
$$

and its Monte Carlo estimator:

$$e\hat{l}e_i = \frac{1}{J}\sum_{j=1}^{J}\left\{\tilde{u}_i^{N(j)} - u^{\dagger}(s_i;\tilde{u}_{i-1}^{N(j)})\right\}. \tag{28}$$

In the following section, simulation studies will be used to compare the distribution of this estimated expected local truncation error between the probailistic state-space solver implemented using an adaptive and an equally spaced design.

4 Results

To illustrate the performance of adaptive step-size selection for the probabilistic numerical solver, we consider two examples. First, we consider the simple second-order initial value problem (Chkrebtii et al. 2016a)

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t^2}u^2 = \sin(2t) - u, \quad t \in [0, 10], \\ \frac{\mathrm{d}}{\mathrm{d}t}u(0) = 0, \;\; u(0) = -1, \end{cases} \tag{29}$$

which is reduced to a first order ODE initial value problem with two states. Figures 2 and 3 show 100 draws from the probabilistic solution in gray with minimum grid sizes $N = 25$ and $N = 100$, respectively. The exact solution $u(t) = u(0)\cos(t) + \left(\frac{\mathrm{d}}{\mathrm{d}t}u(0) + \frac{2}{3}\right)\sin(t) - \sin(2t)$ is superimposed in red for comparison. Note that the second state (lower panel) is the derivative of the first state (upper panel). The gray vertical lines illustrate the density of the adaptively chosen discretization grid. Figure 2 illustrates that larger steps are taken when the derivative is changing quickly, corresponding to large posterior variance in those regions. This is due to the fact that the step length is chosen to maximize the weighted difference between the predictive mean and the vector field evaluation at each candidate point. Figure 3 illustrates that the adaptive algorithm behaves similarly given a finer grid of discretization points.

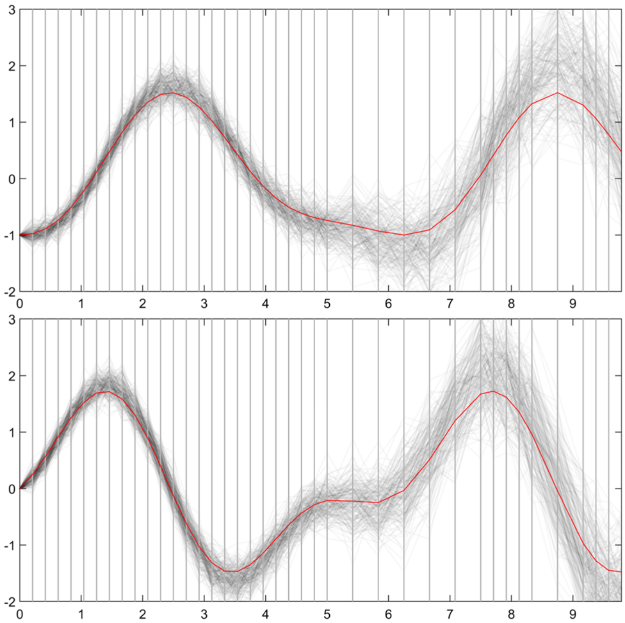

Fig. 2 Marginal sample paths (light solid lines) over the state (top panel) and derivative (bottom panel) trajectories for the second-order ODE initial value problem (29) obtained using algorithm 2 with a minimum grid size of $N = 25$. The exact solution and its derivative are shown as dark solid lines, and discretization grid point locations are identified by vertical lines

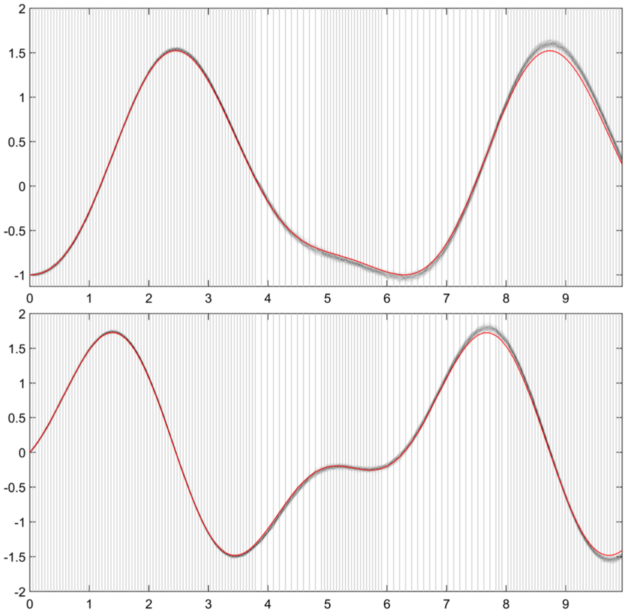

Fig. 3 Marginal sample paths (light solid lines) over the state (top panel) and derivative (bottom panel) trajectories for the second-order ODE initial value problem (29) obtained using Algorithm 2 with a minimum grid size of $N = 100$. The exact solution and its derivative are shown as dark solid lines, and discretization grid point locations are identified by vertical lines, so that a darker region corresponds to a sequence of smaller steps

The second example we consider is the "Lorenz63" model of simplified convection dynamics for three states (Lorenz 1963) on the interval $t \in [0, 5]$. The forward model is defined implicitly by:

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}u^{(1)} = -\sigma(u^{(1)} + u^{(2)}), \\ \frac{\mathrm{d}}{\mathrm{d}t}u^{(2)} = -ru^{(1)} - u^{(2)} - u^{(1)}u^{(3)}, \\ \frac{\mathrm{d}}{\mathrm{d}t}u^{(3)} = u^{(1)}u^{(2)} - bu^{(3)}, \\ u(0) = (-12, -5, 38), \end{cases} \tag{30}$$

with parameter values $(\sigma, r, b) = (10, 8/3, 28)$. This model is stiff due to the presence of linear and nonlinear terms and is therefore challenging to approximate numerically without adaptively selecting the step length. Since the exact solution is unknown, we will use the MATLAB

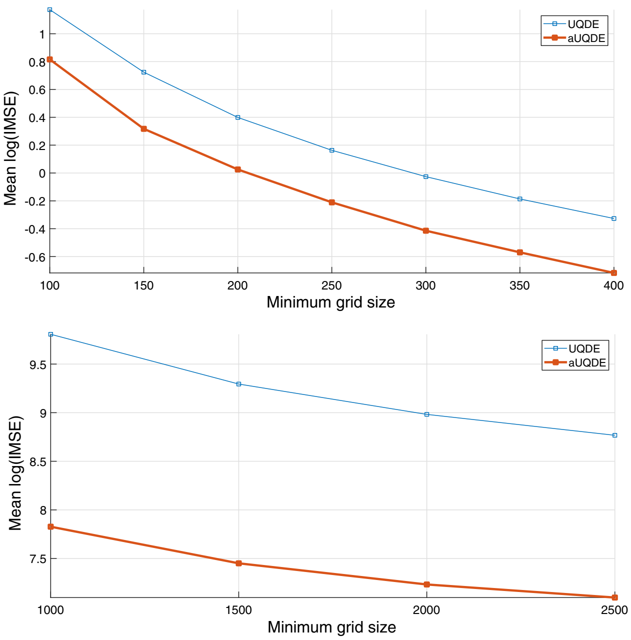

Fig. 4 Mean over 100 simulation runs of the logarithm of integrated mean squared error for the adaptive (aUQDES, thick line) and equally spaced grid probabilistic numerical solver (UQDES, thin line) for two forward models. Top: the ODE initial value problem (29). Bottom: the Lorenz63 model

`ode15` solver for assessing the performance of our adaptive method.

For both systems, we conduct simulations to compare the performance between an adaptive grid probabilistic solver with a probabilistic solver set up over a grid of equally spaced design points. To ensure a fair comparison, we run the adaptive probabilistic solver first. The resulting total number of steps selected by this algorithm is then used to set up the equidistant design for the probabilistic solver implementation. Posterior draws from both implementations are compared in terms of the log integrated mean squared error (IMSE) centered at the exact solution, if available, and at a fine-grid numerical solution otherwise. The IMSE may be viewed as the probabilistic analog of global error, which is in general not accessible in practice and is therefore not directly optimized. However, for stable systems, controlling local error at each step can lead to lower global error or, as in this case, lower IMSE.

Figure 4 illustrates the increased accuracy of using an adaptive UQDE algorithm compared to one that is constructed on an equally spaced grid. Accuracy is compared via the log IMSE over a number of grid sizes for the two test models described above. As expected, the log IMSE decreases as the discretization grid becomes finer, and the adaptive algorithm appears to have greater accuracy than the algorithm implemented over a uniform grid.

Figure 5 is a visualization of the distribution of local errors between the state-space probabilistic solver with adaptive and equally spaced designs. The probabilistic analog of local error described in Sect. 3.3 is estimated at each step of the algorithm for different values $h_{\max}$. Local truncation errors typically increase with the step length, and thus we expect more variability in the local errors for the adaptive algorithm, because step sizes can change. Despite this fact, we observe that the estimated local errors have more mass near zero than the solver based on a uniform design and the overall spread is quite low. The adaptive algorithm appears to out-perform the algorithm without step-size adaptation for all values of the maximum step length considered.

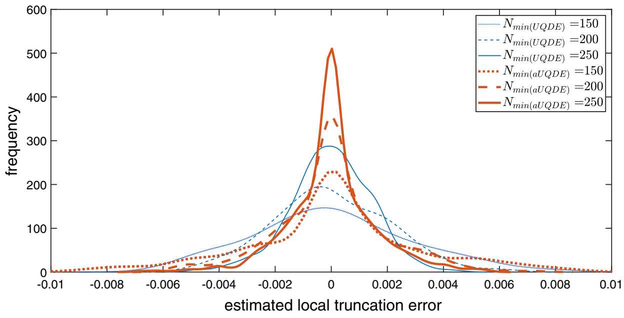

Fig. 5 Kernel density estimate of the distribution of estimated local truncation errors for model (29) under an equally spaced design (thin lines) and an adaptive design (thick lines). As the minimum number of grid points increases (from dotted to solid lines), both densities become more concentrated around 0, as expected. However, the density of estimated local errors appears to always have more mass around zero for the adaptive algorithm as compared to the uniform design

5 Conclusions

The explicit solution to an ODE initial value problem is a source of model uncertainty when the implicitly defined forward model cannot be solved analytically. In this paper, we consider uncertainty quantification based on the state-space probabilistic solver from Chkrebtii et al. (2016a). Viewed as a Kalman filter, this approach can be set up to adaptively select the step length after individual updates. By sequentially optimizing a criterion inspired by principles of experimental design with an information-theoretic utility function, we have developed a technique that decreases the step length when the predicted trajectory differs substantially from the evaluation of the forward model. In order to study and assess the performance of this adaptive algorithm, we proposed an analog of numerical local truncation error for probabilistic solvers. Another proposed way to assess accuracy, the IMSE, is an analog of global error. However, just like global error, it cannot be computed exactly unless the solution is known or can be approximated to a high degree of accuracy, which is the setting for our numerical experiments.

The feature of a probabilistic solver that makes sequential design very appealing is the ability to generate "data" in less time than it takes to complete a single model update. Therefore, the design criterion used can simply be optimized over candidate observations, thus extracting more information from the model. Though this is not the typical scenario when dealing with observational data, other applications that have this feature include design for computer experiments, when the problem is formulated sequentially, such as in Higdon et al. (2013).

Practical consideration in the implementation of adaptive numerical solvers was discussed, including prior specification. A method that chooses step sizes adaptively should also have flexibility in how each of those evaluations is weighted. In other words, the length-scale of the covariance structure must be able to adapt to local changes in the step length. There has been some work on developing such non-stationary covariances for GP models (Paciorek and Schervish 2004, 2006); however, translating this work to a probabilistic model requires integration or differentiation of these covariances in closed-form, which can be a challenging problem.

Decreasing the computational cost of adaptive algorithms is another important issue worth exploring. For numerical methods, the additional computation required to adaptively select the step length is worthwhile, especially in the case of stiff forward models. A similar argument can be made for probabilistic solvers.

In this paper, we have shown that, just like numerical techniques, probabilistic state-space solvers can benefit from step-size adaptation. As for numerical methods, there are different ways of accomplishing this task. We have chosen a technique based on the principles of experimental design with an information-theoretic utility functions. Another possible approach would be to approximate at each step and control the estimated expected local error. Other interesting questions include investigating and comparing such different approaches, both numerically and theoretically.

Acknowledgements This material was developed, in part, at the *Prob Num 2018* workshop hosted by the Lloyd's Register Foundation programme on Data-Centric Engineering at the Alan Turing Institute, UK, and supported by the National Science Foundation, USA. This material was based upon work partially supported by the National Science Foundation under Grant DMS-1127914 to the Statistical and Applied Mathematical Sciences Institute (SAMSI) and the Isaac Newton Institute for Mathematical Sciences' program on Uncertainty Quantification for Complex Systems: theory and methodologies. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the above-named funding bodies and research institutions.

References

Alexanderian, A., Gloor, P.J., Ghattas, O.: On Bayesian A- and D-optimal experimental designs in infinite dimensions. Bayesian Anal. **11**(3), 671 (2016). https://doi.org/10.1214/15-BA969

Bernardo, J.M., Smith, A.F.M.: Bayesian Theory. Wiley, New York (1994)

Butcher, J.: Numerical Methods for Ordinary Differential Equations. Wiley, Hoboken (2008). https://doi.org/10.1002/9780470753767

Chaloner, K., Verdinelli, I.: Bayesian experimental design: a review. Stat. Sci. **10**, 273 (1995). https://doi.org/10.1214/ss/1177009939

Chkrebtii, O., Campbell, D., Calderhead, B., Girolami, M.: Bayesian solution uncertainty quantification for differential equations. Bayesian Anal. **11**(4), 1239 (2016a). https://doi.org/10.1214/16-BA1017

Chkrebtii, O., Campbell, D., Calderhead, B., Girolami, M.: Rejoinder. Bayesian Anal. **4**(11), 1295 (2016b). https://doi.org/10.1214/16-BA1017REJ

Cockayne, J., Oates, C., Sullivan, T., Girolami, M.: Bayesian probabilistic numerical methods (2017). arXiv:1702.03673

Diaconis, P.: Bayesian Numerical Analysis, pp. 163–175. Springer, New York (1988)

Hennig, P., Hauberg, S.: Probabilistic solutions to differential equations and their application to Riemannian statistics. In: Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS), vol. 33 (2014)

Higdon, D., Gattiker, J., Lawrence, E., Jackson, C., Tobis, M., Pratola, M., Habib, S., Heitmann, K., Price, S.: Computer model calibration using the ensemble Kalman filter. Technometrics **55**(4), 488 (2013). https://doi.org/10.1080/00401706.2013.842936

Jazwinski, A.: Stochastic Processes and Filtering Theory. Mathematics in Science and Engineering. Elsevier Science, Amsterdam (1970). https://doi.org/10.1016/S0076-5392(09)60368-4

Kassam, A., Trefethen, L.: Fourth-order time-stepping for stiff PDEs. SIAM J. Sci. Comput. **26**(4), 1214 (2005). https://doi.org/10.1137/S1064827502410633

Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Stat. **22**, 49 (1951). https://doi.org/10.1214/aoms/1177729694

Kuramoto, Y., Tsuzuki, T.: Persistent propagation of concentration waves in dissipative media far from thermal equilibrium. Prog. Theor. Phys. **55**, 356 (1976). https://doi.org/10.1143/PTP.55.356

Lorenz, E.N.: Deterministic nonperiodic flow. J. Atmos. Sci. **20**, 130 (1963). https://doi.org/10.1175/1520-0469

O'Hagan, A.O.: Some Bayesian numerical analysis. Bayesian Stat. **4**, 345 (1992)

Paciorek, C.J., Schervish, M.J.: In: Thrun, S., Saul, L.K., Schölkopf, B. (eds.) Advances in Neural Information Processing Systems, vol. 16, pp. 273–280. MIT Press, Cambridge (2004)

Paciorek, C., Schervish, M.: Spatial modelling using a new class of non-stationary covariance functions. Environmetrics **17**(5), 483 (2006). https://doi.org/10.1002/env.785

Schober, M., Duvenaud, D.K., Hennig, P.: In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K. (eds.) Advances in Neural Information Processing Systems, vol. 27, pp. 739–747. Curran Associates, Inc. Red Hook (2014)

Sebastiani, P., Wynn, H.: Maximum entropy sampling and optimal Bayesian experimental design. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **62**(1), 145 (2000). https://doi.org/10.1111/1467-9868.00225

Shampine, L.F.: Error estimation and control for ODEs. J. Sci. Comput. **25**(1), 3 (2005). https://doi.org/10.1007/BF02728979

Skilling, J.: Bayesian Solution of Ordinary Differential Equations, pp. 23–37. Kluwer Academic Publishers, Seattle (1991)

Stuart, A.M.: Inverse problems: a Bayesian perspective. Acta Numer. **19**, 451 (2010). https://doi.org/10.1017/S0962492910000061

Wang, J., Cockayne, J., Oates, C.: On the Bayesian solution of differential equations (2018). arXiv:1805.07109

Wikle, C.K., Berliner, L.M.: A Bayesian tutorial for data assimilation. Physica D **230**(1), 1 (2007). https://doi.org/10.1016/j.physd.2006.09.017

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.