Hindawi
Journal of Applied Mathematics
Volume 2023, Article ID 7909668, 20 pages
https://doi.org/10.1155/2023/7909668

Research Article

Estimation of Spatially Varying Parameters with Application to Hyperbolic SPDES

David Angwenyi

Masinde Muliro University of Science and Technology, Kenya

Correspondence should be addressed to David Angwenyi; dangwenyi@mmust.ac.ke

Received 5 October 2022; Revised 30 October 2022; Accepted 1 November 2022; Published 10 January 2023

Academic Editor: Sheng Zhang

Parameter estimation is a growing area of interest in statistical signal processing. Some parameters in real-life applications vary in space as opposed to those that are static. Most common methods in estimating parameters involve solving an optimization problem where the cost function is assembled variously, for example, maximum likelihood and maximum a posteriori methods. However, these methods do not have exact solutions to most real-life problems. It is for this reason that Monte Carlo methods are preferred. In this paper, we treat the estimation of parameters which vary with space. We use the Metropolis-Hastings algorithm as a selection criterion for the maximum filter likelihood. Comparisons are made with the use of joint estimation of both the spatially varying parameters and the state. We illustrate the procedures employed in this paper by means of two hyperbolic SPDEs: the advection and the wave equation. The Metropolis-Hastings procedure registers better estimates.

1. Introduction

Most state-space models are characterized by, among other things, parameters—which can be constant or varying. A parameter is comprehended and signified as a measurable factor which defines a model and influences its operation. As the parameter changes, so does the model; expressed differently, a parameter is unique to a model it characterizes. The choice of a certain model, therefore, is achieved by choosing the right parameters. It occurs more often than not—in hidden Markov models, for instance—that measurements are available but the underlying signal is not readily apparent. This forms an example where parameter estimation is paramount: measurements are used to learn the model parameters, which, in turn, are used to fit the model.

Consider a signal and measurement equations, with a spatially varying parameter signified by a d-dimensional vector, θ,

$$\text{Signal}: dx_t = f(x_t, \theta)dt + g(x_t)d\beta_t\,; t_0 \le t, \tag{1a}$$

$$\text{Measurement}: dy_t = h(x_t, \theta)dt + R^{1/2}(t)d\eta_t\,; t_0 \le t, \tag{1b}$$

where the terms are as described in Table 1.

Parameter estimation problem concerns finding the optimal parameter so that the signal best fits the data [1, 2]. This is, classically, achieved by an optimization procedure where a cost function is minimized [3]. The cost function mostly defines the discrepancy between the state and the measurements. Intuitively, parameter estimation can be seen as a procedure for seeking a parameter value that gives the least discrepancy between the state and the corresponding measurements (also known as the algebraic distance or the residual). The method of least squares has been extensively used to define an objective function. Given the increment in measurement, dy_t, of the state, x_t, at time t, the objective function, $\mathcal{J}(\theta)$, in the least-squares sense, is given by

$$\mathcal{J}(\theta) = \int_{t_0}^{t} w_t \| dy_t - h(x_t, \theta)dt \|^2, \tag{2}$$

where w_t is the weighting function.

Table 1: Terms, their descriptions, and dimensions.

Term	Name	Dimension
x_t	State vector	$n \times 1$
$f(x_t, t)$	Drift function	$n \times 1$
$g(x_t, t)$	Diffusion function	$n \times m$
$\{\beta_t, t > t_0\}$	Brownian motion process	$m \times 1$
y_t	Output vector	$r \times 1$
$h(x_t, t)$	Sensor function	$r \times 1$
$R(t)$	Time-function matrix	$r \times r$
$\{\eta_t, t > t_0\}$	Standard Brownian motion process	$r \times 1$

Most commonly used procedures in the framework of least squares include the following: linear least squares, orthogonal least squares, gradient-weighted least squares, and bias-corrected renormalization. This paper, however, attends not to the study of least-squares approaches, they being outside the scope of its design. Suffice it to only direct the interested reader to the article [4] for an elaborate explanation and application of least-squares methods in computer vision. Instead, we study parameter estimation by means of filtering. But before that, we mention a few merits and demerits of least squares and other methods defined by algebraic distances.

The use of algebraic distances in defining a cost function is computationally efficient, and closed-form solutions are possible. The end result, however, is not satisfactory. This is due, in one part, to the fact that the objective function is mostly not invariant with respect to Euclidean transformations, for example, translation. This limits the coordinate systems to be used. In the other part, outliers may not contribute to the parameters the same way as inliers [4]. Other more satisfactory parameter estimation methodologies are highly desired. We consider, in this paper, the use of Bayesian inference techniques.

Estimation of parameters by means of a filter can be achieved in a number of ways, one of them being the use of the filter evidence, or its near approximation, and selection criteria for parameters which give a reasonable estimate of the evidence. The second method involves updating the parameters and the state at the same time. This is known as dual estimation, which further subdivides into joint estimation and a dual filter. Joint estimation entails subjoining the vector of parameters to the state vector to form an extended state-space. The filter is then implemented and run forward in time with the hope of filter convergence to the optimal state and parameter values. A dual filter, on the other hand, involves implementing a filter for the state and parameters simultaneously. The filter provides a self-correcting mechanism which may lead to convergence of state and parameter estimates.

The rest of this paper is arranged as follows. In the first place, we introduce some notions on Bayesian inference of parameters where we pass the limit in the mean in order to move from a discrete setting to a continuum in time. We introduce the different ways in which a filter can be used for parameter estimation. In anticipation of application in the later part of the paper, we introduce the stochastic advection and wave equation together with their spatial discretisation. We then illustrate how parameters are to be estimated by means of a filter likelihood and the dual filters. Results and discussions follow, and the conclusion forms the closing part of this paper.

2. Bayesian Parameter Inference

In Bayesian inference of parameters, the parameters are treated as a random variable. The parameter is assigned a prior, $\pi_{t_0}(\theta)$, based on some initial belief. Let t_n such that $t_{n+1} > t_n \forall n = 0, 1, 2, \cdots N$ be a partition of the interval $[t_0, T]$ and let $\delta t = t_{n+1} - t_n$. Bayes' rule gives the joint posterior of parameters and the states;

$$\begin{aligned}
&\pi_{[t_0,T]}(x, \theta | Y_T) \\
&\quad \approx \underset{\substack{\delta t \longrightarrow 0 \\ N \longrightarrow \infty}}{\text{l.i.m.}} \pi_{t_0:t_N}\left(x_{t_0:t_N}, \theta | y_{t_0:t_N}\right) \\
&\quad = \underset{\substack{\delta t \longrightarrow 0 \\ N \longrightarrow \infty}}{\text{l.i.m.}} \frac{\pi_{t_0:t_N}\left(y_{t_0:t_N} \middle| x_{t_0:t_N}, \theta\right) \pi_{t_0:t_N}\left(x_{t_0:t_N} | \theta\right) \pi_{t_0}(\theta)}{\pi_{t_0:t_N}\left(y_{t_0:t_N}\right)},
\end{aligned} \tag{3a}$$

where $Y_T = y_{[t_0,T]}$,

$$\begin{aligned}
&\pi_{t_0:t_N}\left(y_{t_0:t_N} \middle| x_{t_0:t_N}, \theta\right) = \prod_{n=1}^{N} \pi_{t_n}\left(y_{t_n} \middle| x_{t_n}, \theta\right), \\
&\pi_{t_0:t_N}\left(x_{t_0:t_N} | \theta\right) = \pi_{t_0}\left(x_{t_0} | \theta\right) \prod_{n=1}^{N} \pi_{t_n}\left(x_{t_n} | x_{t_{n-1}}, \theta\right).
\end{aligned} \tag{3b}$$

Now to arrive at the marginal posterior of parameters, we integrate out the states from the joint posterior of states and parameters, Equation (3a):

$$\begin{aligned}
&\pi_{t_0:t_N}\left(\theta | y_{t_0:t_N}\right) \\
&\quad = \int \frac{\pi_{t_0:t_N}\left(y_{t_0:t_N} \middle| x_{t_0:t_N}, \theta\right) \pi_{t_0:t_N}\left(x_{t_0:t_N} | \theta\right) \pi_{t_0}(\theta)}{\pi_{t_0:t_N}\left(y_{t_0:t_N}\right)} dx_{t_0:t_N}.
\end{aligned} \tag{4}$$

It turns out that direct computation of the integral in (4) is difficult, especially with the increase in measurements [2]. This challenge is circumvented through the use of recursive techniques which include the use of filters and smoothers, maximum a posteriori (MAP) estimates, and drawing samples from the posterior using Markov Chain Monte Carlo (MCMC) methods.

To use recursive methods, we begin with the following parameter posterior

$$\pi(\theta|Y_T) \approx \underset{\substack{\delta t \longrightarrow 0 \\ N \longrightarrow \infty}}{\text{l.i.m.}}\, \pi\left(\theta|y_{t_0:t_N}\right) \propto \underset{\substack{\delta t \longrightarrow 0 \\ N \longrightarrow \infty}}{\text{l.i.m.}}\, \pi_{t_0:t_N}\left(y_{t_0:t_N}\middle|\theta\right)\pi_{t_0}(\theta), \tag{5a}$$

where

$$\begin{aligned}\pi_{t_0:t_N}\left(y_{t_0:t_N}\middle|\theta\right) &= \prod_{n=1}^{N} \pi_{t_n}\left(y_{t_n}\middle|y_{t_1:t_{n-1}},\theta\right) \\ &= \prod_{n=1}^{N} \pi_{t_n}\left(y_{t_n}\middle|x_{t_n},\theta\right)\pi_{t_n}\left(x_{t_n}\middle|y_{t_1:t_{n-1}},\theta\right)dx_{t_n}.\end{aligned} \tag{5b}$$

The state's predictive distribution, $\pi_{t_n}(x_{t_n}|y_{t_1:t_{n-1}},\theta)$, can be computed recursively as follows [2]:

$$\pi_{t_n}\left(x_{t_n}\middle|y_{t_1:t_{n-1}},\theta\right) = \int \pi_{t_n}\left(x_{t_n}\middle|x_{t_{n-1}},\theta\right)\pi_{t_{n-1}}\left(x_{t_{n-1}}\middle|y_{t_1:t_{n-1}},\theta\right)dx_{t_{n-1}}. \tag{6a}$$

Instead of working with the posterior, $\pi(\theta|Y_T)$, it is quite convenient to use the negative log-posterior obtained by expressing the posterior as follows:

$$\pi(\theta|Y_T) \approx \underset{\substack{\delta t \longrightarrow 0 \\ N \longrightarrow \infty}}{\text{l.i.m.}}\, \pi\left(\theta|y_{t_0:t_N}\right) \propto \underset{\substack{\delta t \longrightarrow 0 \\ N \longrightarrow \infty}}{\text{l.i.m.}}\, \exp\left(-\psi_T(\theta)\right), \tag{7}$$

where the *energy function*, $\psi_T(\theta)$, is given by

$$\psi_T(\theta) = -\log\left(\pi_{t_0:t_N}\left(y_{t_0:t_N}\middle|\theta\right)\right) - \log\left(\pi_{t_0}(\theta)\right). \tag{8}$$

The maximum a posteriori (MAP) estimate can then be obtained by the mode of the posterior distribution or, equivalently, the minimum of the energy function, the latter of which is easier to compute; that is,

$$\widehat{\theta}_{\text{MAP}} = \underset{\theta}{\text{argmax}}\,\pi(\theta|Y_T) = \underset{\theta}{\text{argmin}}\,\psi_T(\theta). \tag{9}$$

One demerit of the MAP estimate is that it yields a point estimate of the parameter posterior and therefore ignores the dispersion of the estimate. Setting the prior, $\pi_{t_0}(\theta)$, to be a uniform density, (9) yields a maximum likelihood estimate.

3. Metropolis-Hastings Method

Metropolis-Hastings (named after Nicholas Constantine Metropolis (1915-1999) and Wilfred Keith Hastings (1930-2016)) [5] is a Markov Chain Monte Carlo sampling algorithm with optimal convergence. It is premised on detailed balance and ergodicity. Given a probability density, say $\pi(\theta)$, from which it is difficult to sample (for instance, if the said distribution is known to a normalization constant), and given another distribution $\rho(\theta)$, say from which it is easy to sample, then detailed balance is the condition

$$\pi(\theta_k)\rho(\theta_k|\theta_{k+1}) = \pi(\theta_{k+1})\rho(\theta_{k+1}|\theta_k), \tag{10}$$

where $\rho(\theta_k|\theta_{k+1})$ is a transition distribution. The detailed balance condition is necessary for any random walk to asymptotically converge to a stationary distribution. Ergodicity is meant that there is a nonzero probability in moving from a state to any other state in a Markov Chain.

The following algorithm summarises the Metropolis-Hastings procedure.

4. Dual Estimation

Dual estimation comprehends simultaneous estimation of state and parameters by means of an appropriate filter. The self-correcting mechanism of the filter is taken advantage of to converge to both the true state and the true parameters. Depending on the initial parameter, the filter sooner or later converges to the true parameter value. Dual estimation can be achieved in two ways: joint estimation and by a dual filter [6–8].

4.1. Joint Estimation (Augmented State-Space). In joint estimation, the state vector is augmented with the vector of parameters to form an extended state-space, and then, the filter is run forward in time for an update of both the state and the parameters. The parameters are induced with artificial dynamics or are made to assume a random walk; that is, respectively,

$$dz_t = \zeta_t\,;\, t_0 \le t, \tag{11}$$

where

$$dz_t = \begin{pmatrix} dx_t \\ d\theta_t \end{pmatrix}, \zeta_t = \begin{pmatrix} f(x_t,\theta)dt + g(x_t)d\beta_t \\ 0 \end{pmatrix}, \tag{12a}$$

or where

$$\zeta_t = \begin{pmatrix} f(x_t,\theta)dt + g(x_t)d\beta_t \\ \sigma d\chi_t \end{pmatrix}, \tag{12b}$$

in which $\{\chi_t, t > t_0\}$ is a d-dimensional standard Brownian motion vector process and σ is a small constant. A filter is then implemented with the augmented state z_t in the place of x_t. The demerit of this method is that the extended state-space has an increased degree of freedom owing to many unknowns, of both the state and the parameters, which renders the filter unstable and intractable, especially in nonlinear models [8].

4.2. Dual Filter. Dual filtering of the state and parameters is attained by use of two filters, one for state update and another for updating parameters, both run simultaneously. The two filters interact symbiotically in that one provides the update of the state to be used by the other, whilst the

other provides an update of the parameters to be used by the former. A very good example in literature is the dual extended Kalman filter [9] used for estimating neural network models and the weights. In this case, the state is the model signal and the weights are parameters. Another example appears in [10] where a dual filter comprising the ensemble transform particle filter (ETPF) and the feedback particle filter (FPF) is used for simultaneous estimation of the state of a wave equation and its speed.

The model for the dual filter of our consideration comprises a d-dimensional vector equation of artificial dynamics of parameters together with the state-space model, Equations (1a) and (1b):

$$\text{Parameter} : d\theta_t = 0, t_0 \le t, \tag{13a}$$

$$\text{Signal} : dx_t = f(x_t, \theta_t)dt + g(x_t)d\beta_t, t_0 \le t, \tag{13b}$$

$$\text{Measurement} : dy_t = h(x_t)dt + R^{1/2}(t)d\eta_t, t_0 \le t, \tag{13c}$$

where the nomenclature and dimensions remain as stipulated for Equations (1a) and (1b).

In the following, we introduce the equations to which we shall apply dual filters in estimating spatially varying parameters.

5. Application Equations

5.1. Advection Equation. We take up an advection equation, excited with the space-time white noise process, with some diffusion term added to it, and on a periodic spatial domain of length L, which we write as follows:

$$\frac{\partial u}{\partial t} = \frac{\partial (C(x)u)}{\partial x} + \mu \frac{\partial^2 u}{\partial x^2} + \sigma \dot{\beta}_{x,t}, 0 \le t \le T_t \times 0 \le x \le L, \tag{14}$$

where $C(x)$ is a spatially varying velocity (of which constant velocity is a special case), σ is a constant, and $u(x,t)$ is the function to be obtained, whose function describes the state of the signal. μ is a constant whilst $\dot{\beta}_{x,t}$ is the space-time white noise process where the dot denotes the singularity of the noise process.

Equation (14) needs a remark owing to the roughness of the stochastic-forcing term $\dot{\beta}_{x,t}$, which is a mixed distributional derivative of the Brownian sheet. As is well known (see [11] for details), the Brownian sheet is nowhere differentiable. We, however, use the method introduced in [12]; that is, we approximate the noise term as follows. Let the domain $0 \le t \le T_t \times 0 \le x \le L$ be tessellated into rectangles $[t_n, t_{n+1}] \times [x_i, x_{i+1}]$ of dimensions $\delta t \times \delta x$ each, for $n = 1, 2, 3, \cdots, T$ and $i = 1, 2, 3, \cdots, N$ so that $\delta t = T_t/T$ and $\delta x = L/N$. Then,

$$\dot{\beta}_{x,t} \approx \frac{1}{\delta x \delta t} \sum_{i=1}^{N} \sum_{n=1}^{T} \omega_{i,n} (\delta x \delta t)^{1/2} \chi_i(x) \chi_n(t), \tag{15}$$

where $\{\omega_{i,n}\}_{i=1}^{N}$ are independent and identically distributed random variables of mean 0 and unit variance. $\chi_i(x)$ and $\chi_n(t)$ are characteristic functions and are given by

$$\chi_n(t) = \begin{cases} 1, & \text{if } t_n \le t \le t_{n+1}, \\ 0, & \text{otherwise}, \end{cases}$$
$$\chi_i(x) = \begin{cases} 1, & \text{if } x_i \le x \le x_{i+1}, \\ 0, & \text{otherwise}. \end{cases} \tag{16}$$

By a three-point upwind scheme in space [13], we discretise (14) and arrive at the following:

$$\frac{du_i}{dt} \approx \frac{3C_i u_i - 4C_{i-1}u_{i-1} + C_{i-2}u_{i-2}}{2\delta x} + \mu \frac{3u_{i+2} - 16u_{i+1} + 26u_i - 16u_{i-1} + 3u_{i-2}}{4\delta x^2} + \sigma \frac{1}{\sqrt{\delta x}} \dot{\omega}_i, \tag{17}$$

where δx is the spatial step size and $\{\omega_{i,t}, t > t_0\}$ is the standard Brownian motion process. The ith grid point is represented by $x_i = i\delta x$. With this notation, $u_{i,n}$ is understood to mean the value of u at the ith grid point at time t_n.

Time discretisation, by means of the Euler-Maruyama scheme, leads to

$$u_{i,t_{n+1}} = u_{i,t_n} + \frac{3C_i u_{i,t_n} - 4C_{i-1}u_{i-1,t_n} + C_{i-2}u_{i-2,t_n}}{2\delta x} \delta t + \mu \frac{3u_{i+2,t_n} - 16u_{i+1,t_n} + 26u_{i,t_n} - 16u_{i-1,t_n} + 3u_{i-2,t_n}}{4\delta x^2} \delta t + \sigma \frac{\delta t^{1/2}}{\sqrt{\delta x}} \omega_{i,t_n}, \tag{18}$$

where ω_{i,t_n} is a random variable of mean 0 and variance 1. The time increment, $\delta t > 0$, is such that the limit of δu_i as $\delta t \longrightarrow 0$ is du_i. Furthermore, $n = 1, 2, 3, \cdots, T$. We use the following initial value.

$$u(x, t_0) = \sin(x). \tag{19}$$

Moreover, $C(x) = e^{\lambda(x)}$ where

$$\lambda(x) = \sin(2\pi x). \tag{20}$$

Considering every grid point in (18) leads to a vector representation of the signal u. To do so requires the following shorthand for operations:

$$(D_1 u)_i \coloneqq \frac{3u_i - 4u_{i-1} + u_{i-2}}{2\delta x}, \quad \forall i = 1, 2, 3, \cdots, N, \tag{21}$$

$$\left(D_1^T u\right)_i \coloneqq \frac{u_{i+2} - 4u_{i+1} + 3u_i}{2\delta x}, \quad \forall i = 1, 2, 3, \cdots, N, \tag{22}$$

so that

$$\begin{aligned}\left(D_1 D_1^T u\right)_i &\coloneqq \frac{3u_{i+2} - 12u_{i+1} + 9u_i}{4\delta x^2} + \frac{-4u_{i+1} + 16u_i - 12u_{i-1}}{4\delta x^2} \\ &\quad + \frac{u_i - 4u_{i-1} + 3u_{i-2}}{4\delta x^2} \\ &= \frac{3u_{i+2} - 16u_{i+1} + 26u_i - 16u_{i-1} + 3u_{i-2}}{4\delta x^2}, \quad \forall i = 1, 2, 3, \cdots, N.\end{aligned} \tag{23}$$

We finally have

$$u_{t_{n+1}} = u_{t_n} + F(t_n)u_{t_n}\delta t + G(t_n)\omega_{t_n}, \tag{24}$$

where u_{t_n} is an N-dimensional column vector at time t_n comprising of elements $u_{i,t_n}, i = 1, 2, 3, \cdots, N$, and

$$F(t_n) = \left[D_1 C_{\text{diag}} - \mu D_1 D_1^T\right], G(t_n) = \left[\sigma \frac{\delta t^{1/2}}{\sqrt{\delta x}} I_{N\times N}\right], \tag{25}$$

in which C_{diag} is a diagonal matrix made of the elements of C and $I_{N\times N}$ is the Nth-order identity matrix.

The surface and contour plots for the stochastic advection equation are shown below, that is, when $\sigma = 0.1$. The ruggedness evident in Figure 1 is consequent upon the addition of noise to the underlying dynamics.

In the next subsection, we introduce the wave equation.

5.2. Wave Equation. The wave equation—for our consideration—is given by

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial(C(x)\partial u/\partial x)}{\partial x} + \mu \frac{\partial^3 u}{\partial x^2 \partial t} + \sigma \dot{\beta}_{x,t}, \quad 0 \le t \le T_t \times 0 \le x \le L, \tag{26}$$

where $C(x) = e^{\lambda(x)}$ is the wave velocity and is for a wave travelling in a heterogeneous medium and $u(x, t)$ is the function to be obtained, whose function, as in the previous subsection, describes the state of the signal. Moreover, σ is a constant and $\dot{\beta}_{x,t}$, as before, is the space-time white noise.

We employ mixed difference schemes to discretise (26) in space, so that we have

$$\frac{du_i}{dt} \approx p_i, \tag{27a}$$

$$\frac{dp_i}{dt} \approx \frac{C_{i+1}w_{i+1} - C_i w_i}{\delta x} + \mu \frac{p_{i+1} - 2p_i + p_{i-1}}{\delta x^2} + \frac{1}{\sqrt{\delta x}}\dot{\omega}_i, \tag{27b}$$

where $w_i \coloneqq (u_i - u_{i-1})/\delta x$ and δx is the spatial step size. For time integration, we use Verlet's method, because of its geometric properties, namely, volume preservation, symplecticity, conservation of first integrals, and reversibility [14, 15]—whose method, applied to the deterministic part of Equations (27a) and (27b), yields

$$u_{i,t_{n+1}} = u_{i,t_n} + p_{i,t_{n+1/2}}\delta t, \tag{28a}$$

$$\begin{aligned}p_{i,t_{n+1/2}} &= p_{i,t_n} + \frac{C_{i+1}w_{i+1,t_n} - C_i w_{i,t_n}}{\delta x}\frac{\delta t}{2} \\ &\quad + \mu \frac{p_{i+1,t_n} - 2p_{i,t_n} + p_{i-1,t_n}}{\delta x^2}\frac{\delta t}{2},\end{aligned} \tag{28b}$$

$$\begin{aligned}p_{i,t_{n+1}} &= p_{i,t_{n+1/2}} + \frac{C_{i+1}w_{i+1,t_{n+1}} - C_i w_{i,t_{n+1}}}{\delta x}\frac{\delta t}{2} \\ &\quad + \mu \frac{p_{i+1,t_{n+1}} - 2p_{i,t_{n+1}} + p_{i-1,t_{n+1}}}{\delta x^2}\frac{\delta t}{2} + \sigma \frac{\delta t^{1/2}}{\sqrt{\delta x}}\omega_{i,t_n},\end{aligned} \tag{28c}$$

where δt is the time step and $w_{i,t_n} \coloneqq (u_{i,t_n} - u_{i-1,t_n})/\delta x$. Substituting Equation (28b) into Equations (28a) and (28c), we get

$$\begin{aligned}u_{i,t_{n+1}} &= u_{i,t_n} + p_{i,t_n} + \frac{C_{i+1}w_{i+1,t_n} - C_i w_{i,t_n}}{\delta x}\frac{\delta t}{2} \\ &\quad + \mu \frac{p_{i+1,t_n} - 2p_{i,t_n} + p_{i-1,t_n}}{\delta x^2}\frac{\delta t}{2},\end{aligned} \tag{29a}$$

$$\begin{aligned}p_{i,t_{n+1}} &= p_{i,t_n} + \frac{C_{i+1}w_{i+1,t_n} - C_i w_{i,t_n}}{\delta x}\frac{\delta t}{2} \\ &\quad + \mu \frac{p_{i+1,t_n} - 2p_{i,t_n} + p_{i-1,t_n}}{\delta x^2}\frac{\delta t}{2} \\ &\quad + \frac{C_{i+1}w_{i+1,t_{n+1}} - C_i w_{i,t_{n+1}}}{\delta x}\frac{\delta t}{2} \\ &\quad + \mu \frac{p_{i+1,t_{n+1}} - 2p_{i,t_{n+1}} + p_{i-1,t_{n+1}}}{\delta x^2}\frac{\delta t}{2} + \sigma \frac{\delta t^{1/2}}{\sqrt{\delta x}}\omega_{i,t_n}.\end{aligned} \tag{29b}$$

We use the following initial values:

$$u(x, 0) = \exp\left(-4(x - 0.5L).^2\right), \tag{30a}$$

$$p(x, 0) = 0, \tag{30b}$$

where L is the length of the domain. Now, $C(x) = e^{\lambda(x)}$ where

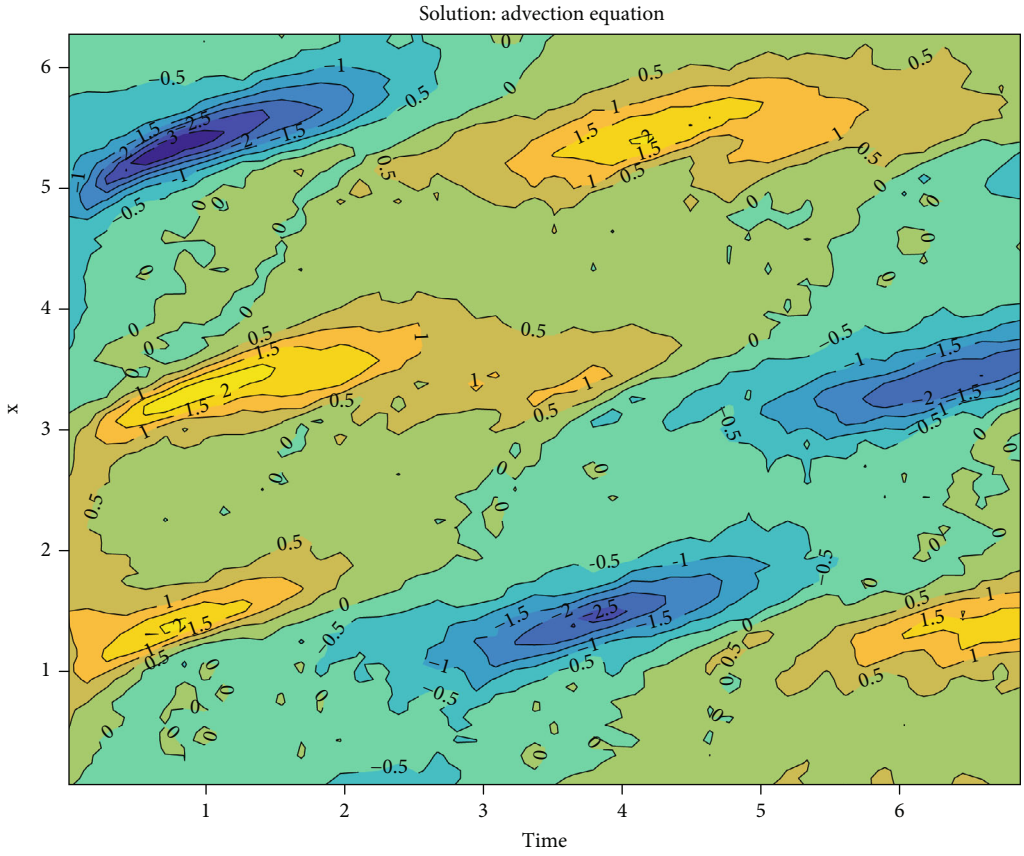
$$\lambda(x) = \sin(x). \tag{31}$$

Considering every grid point leads to a vector representation of the signal u. We use the following shorthand:

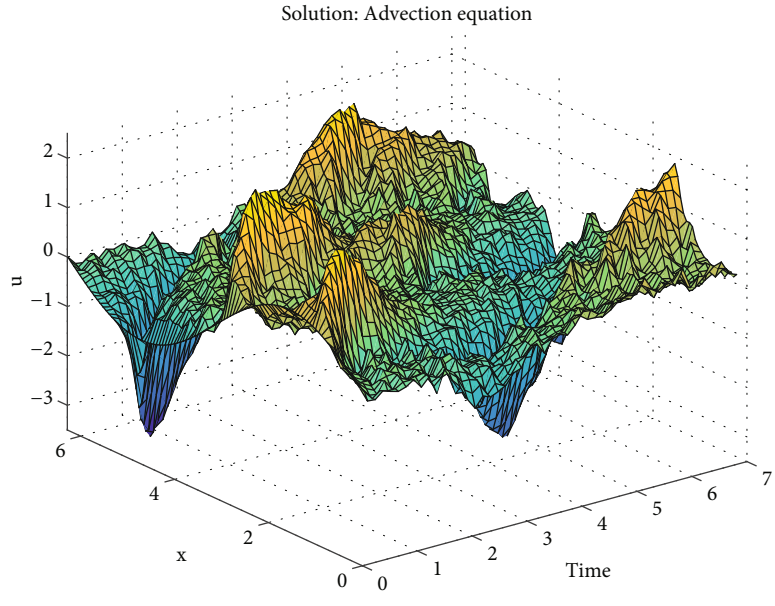
$$\left(D_2 u\right)_i \coloneqq \frac{u_i - u_{i-1}}{\delta x}, \quad \forall i = 1, 2, 3, \cdots, N. \tag{32}$$

Equations (29a) and (29b) then become

$$\underline{u}_{t_{n+1}} = \underline{u}_{t_n} + Ft_n \underline{u}_{t_n} + G(t_n)\underline{\omega}_{t_n}, \tag{33}$$



(a)



(b)

Figure 1: Contour and surface plots for a single realisation of the stochastic advection equation under the following setting: $L = 2\pi$, $N = 100$, $\delta x = L/N$, $\delta t = 0.007$, $T = 1000$, $\sigma = 0.1$, $\mu = 0.01$, $C(x) = e^{\lambda(x)}$ where $\lambda(x) = \sin(2\pi x)$, and $u_{t_0} = \sin(x)$.

where

$$F(t_n) = -I_{2N\times 2N} + \begin{bmatrix} I_{N\times N} - \frac{\delta t^2}{2} D_2\left(C_{\text{diag}}(x) D_2^T\right) & \delta t I_{N\times N} - \frac{\delta t^2}{2}\mu D_2 D_2^T \\ -\frac{\delta t}{2} D_2\left(C_{\text{diag}}(x) D_2^T\right) & I_{N\times N} - \frac{\delta t}{2}\mu D_2 D_2^T \end{bmatrix} \times \begin{bmatrix} I_{N\times N} & 0_{N\times N} \\ \frac{\delta t}{2} D_2\left(C_{\text{diag}}(x) D_2^T\right) & I_{N\times N} - \frac{\delta t}{2}\mu D_2 D_2^T \end{bmatrix}^{-1},$$

$$\underline{u} = \begin{pmatrix} p \\ u \end{pmatrix}, G(t_n) = \begin{bmatrix} 0_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & \sigma \frac{\delta t^{1/2}}{\sqrt{\delta x}} I_{N\times N} \end{bmatrix}, \underline{\omega} = \begin{pmatrix} \omega \\ \omega \end{pmatrix}, \tag{34}$$

where $I_{2N\times 2N}$ is the identity matrix of order $2N$ whilst $0_{N\times N}$ is an Nth-order null matrix. 0_N is an Nth-dimensional null vector. $F(t_n) \in \mathbb{R}^{2N\times 2N}$, $G(t_n) \in \mathbb{R}^{2N\times 2N}$, and $\underline{\omega} \in \mathbb{R}^N$.

The contour and surface plots for a single realisation of stochastic wave equation are in Figure 2.

For the purpose of estimating the speed of the wave, which varies spatially, we give a little prelude to estimation of spatially varying parameters in the next section.

6. Estimating Spatially Varying Parameters

Varying parameters exist naturally, an example being in the speed of a wave moving in a heterogeneous media. Varying parameters present a challenge owing to their varying nature as opposed to static (in time) parameters. There are three broad categories of varying parameters: spatially varying parameters, parameters that vary with time, and parameters that vary with both time and space. Estimation of time-varying parameters, albeit for deterministic models, and application to estimation of parameters in a HIV/AIDS model, appears in [16]—in which least-squares methods have been used. Although consistent estimates are realised, extension to nonlinear models remains to be realised. In this chapter, we study spatially varying parameters using filtering algorithms. We also provide applications in order to evaluate the performance of the algorithms introduced here.

The strategy employed in this study comprises of three steps:

(1) Express the parameter $\theta(x)$ as a Fourier series with a given number of modes, say, N_m and collect all the coefficients in a vector λ

(2) By means of an appropriate filtering algorithm, estimate the vector of hyperparameters, λ

(3) Substitute the estimated constant coefficients back in the Fourier series to obtain an estimate of the parameter $\theta(x)$

To illustrate this method, we employ it to estimate the velocity of a wave travelling in a heterogeneous media. Let the parameter, $C(x)$, $x \in \mathbb{R}^N$, where N is the number of dimensions, be given by

$$C(x) = \exp\left(f(x)\right), \tag{35}$$

where

$$f(x) = a_0 + \sum_{k=1}^{N_m} \frac{a_k}{k^2} \sin(kx) + \sum_{k=1}^{N_m} \frac{b_k}{k^2} \cos(kx), \quad k = 1, 2, 3, \cdots, N_m, \tag{36}$$

where the coefficients a_0, a_k, and b_k are drawn randomly from a normal distribution of mean 0 and variance 1. This way, the parameter $C(x)$ will be positive for all values of x. The aim of this section is to obtain an estimate $C(x)_{est}$ of the wave velocity $C(x)$ by means of the Kalman-Bucy filter (KBF) and ensemble Kalman-Bucy filter (EnKBF) [17]. To this end, we use a function

$$g(x) = A_0 + \sum_{k=1}^{\aleph_m} \frac{A_k}{k^2} \sin(kx) + \sum_{k=1}^{\aleph_m} \frac{B_k}{k^2} \cos(kx), \quad k = 1, 2, 3, \cdots, \aleph_m, \tag{37}$$

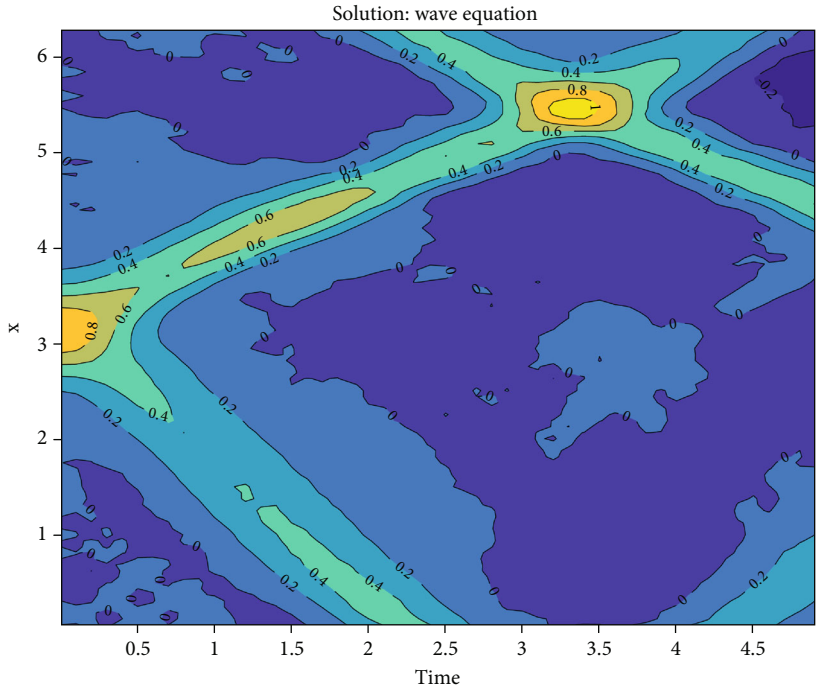
where $\aleph_m \leq N_m$ so that

$$C(x)_{est} = \exp\left(g(x)\right). \tag{38}$$

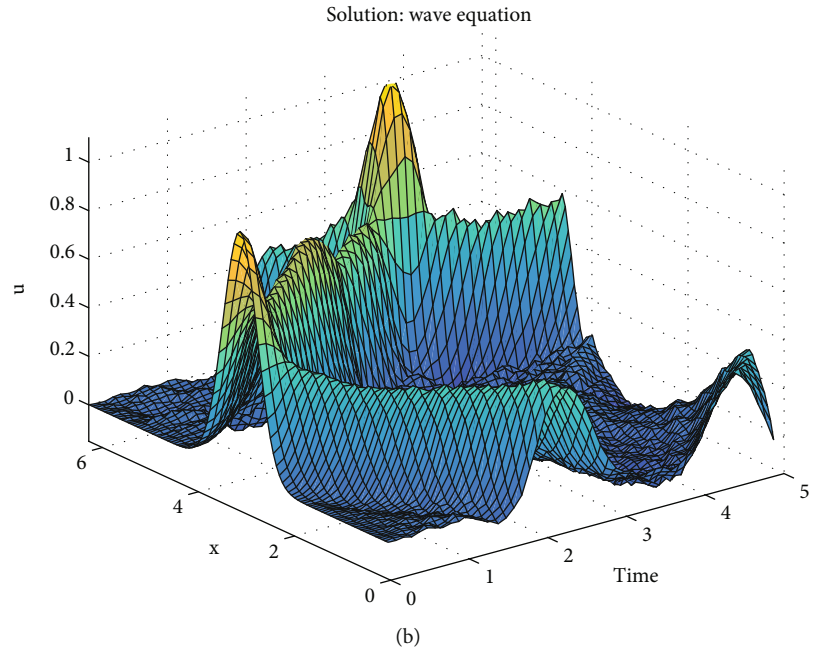
Let $\lambda \in \mathbb{R}^{2\aleph_m+1}$ be a vector whose elements are the coefficients of the function $g(x)$. That is,

$$\lambda = \begin{pmatrix} A_0 \\ A_1 \\ B_1 \\ A_2 \\ B_2 \\ \vdots \\ A_{\aleph_m} \\ B_{\aleph_m} \end{pmatrix}. \tag{39}$$

Estimation of the spatially varying parameter, $C(x)$, is equivalent to estimating the parameters that form the vector λ. We consider two methods: using the likelihood with the Metropolis-Hastings method and using a dual filter. Let us now consider each method in turn.



(a)



(b)

FIGURE 2: Plots for a single realisation of the stochastic wave equation. The settings are $L = 2\pi$, $N = 100$, $\delta x = L/N$, $\delta t = 0.005$, and $T = 1000$, $\mu = 0.01$, $\sigma = 0.2$, $C(x) = e^{\lambda(x)}$ where $\lambda(x) = \sin(x)$, and $u_{t_0} = \exp(-4(x - 0.5L).^2)$.

7. Using the Likelihood with Metropolis-Hastings Method

We now adapt Algorithm 1 to estimating the vector of parameters, λ. We let each parameter λ_i have artificial dynamics with the transition density

$$\zeta_{k+1}\left(\lambda_{i,k+1}\middle|\lambda_{i,k}\right) = \mathcal{N}\left(\lambda_{i,k}\cos(\phi), \frac{\omega}{\aleph}\sin(\phi)\right), \tag{40}$$

where $\aleph$ is the mode number and the constants ϕ and ω are to be chosen. The density π is defined by the filter likelihood,

$$\pi\left(\delta y_{[t_0,T]}\middle|\tilde{u}_{[t_0,T]}, \lambda_k\right) = \underset{\substack{\delta t\longrightarrow 0\\ N\longrightarrow\infty}}{\text{l.i.m.}} \pi_{t_0:t_N}\left(\delta y_{t_0:t_N}\middle|\tilde{u}_{t_0:t_N}, \lambda_k\right), \tag{41}$$

where $\tilde{u}_{t_n}$ is the filter prediction of the state at time t_n. Notice that the parameters, λ, are implicitly contained in the dynamics.

1: Draw $\widetilde{\theta}$ from $\rho(\widetilde{\theta}|\theta_k)$
2: Set $\theta_{k+1} \longleftarrow \widetilde{\theta}$ with probability $\alpha = \min\left(1, (\pi(\widetilde{\theta})\rho(\theta_k|\widetilde{\theta})/\pi(\theta_k)\rho(\widetilde{\theta}|\theta_k))\right)$
3: Otherwise, set $\theta_{k+1} \longleftarrow \theta_k$

Algorithm 1: Metropolis-Hastings.

Require: δt, $\aleph_m$, N, u_{t_0}, π_{t_n}, and λ_k.
Ensure: $\{\lambda_k\}_{k=1}^{T}$.
1: **for** $k = 1$ **to** N **do**
2: Draw $\tilde{\lambda} \sim \mathcal{N}(\lambda_{i,k}\cos(\phi), (\omega/\aleph_m)\sin(\phi))$ $\forall i = 1, 2, \cdots, 2\aleph_m + 1$.
3: Compute $C_k(x) = \exp(g(x, \lambda_k))$.
4: **for** $n = 1$ **to** T, $\delta t > 0$ **do**
5: Run a single-step KBF prediction mean $\tilde{u}_{t_{n+1}} = u_{t_n} + F(t_n, \lambda_k)u_{t_n}\delta t$
6: Run a single-step KBF prediction covariance $\tilde{P}_{t_{n+1}} = P_{t_n} + F(t_n, \lambda_k)P_{t_n}\delta t + P_{t_n}F^T(t_n, \lambda_k)\delta t + G(t_n)G^T(t_n)\delta t$
7: Run a single-step KBF analysis mean $u_{t_{n+1}} = \tilde{u}_{t_{n+1}} + P_{t_n}H^T(t_n)R^{-1}(t_n)(dy_{t_n} - H(t_n, \lambda_k)\tilde{u}_{t_n}\delta t)$
8: Run a single-step KBF analysis covariance $P_{t_{n+1}} = \tilde{P}_{t_{n+1}} + G(t_n)G^T(t_n)\delta t - \tilde{P}_{t_n}H^T(t_n)R^{-1}(t_n)H(t_n)P_{t_n}\delta t$
9: **end for**
10: Metropolis-Hastings
11: Compute $\alpha_{\text{ratio}} = (\pi(\delta y_{t_N:t_0}|\tilde{u}_{t_N}, \lambda_k)/\pi(\delta y_{t_N:t_0}|\tilde{u}_{t_N}, \lambda_{k-1}))(\rho_\lambda(\tilde{\lambda})/\rho_\lambda(\lambda_k))$
12: Compute $\alpha = \min(1, \alpha_{\text{ratio}})$
13: **if** $\alpha > U(0, 1)$ **then**
14: $\tilde{\lambda} = \lambda_k$
15: **else**
16: $\lambda_k = \tilde{\lambda}$
17: **end if**
18: **end for**

Algorithm 2: KBF likelihood with MH.

Example 1. Advection equation.

We take the advection equation, of Section 5.1, and the given initial conditions. Furthermore, let there be time-continuous measurements of the state, u, given by

$$dy_t = H(t)u(x, t)dt + Q(t)d\eta_{x,t}, \tag{42}$$

where $\{\eta_{t,x}\}$ is the standard space-time Brownian motion process. The initial values of u_t, $\{\beta_t\}$, and $\{\eta_{t,x}\}$ are uncorrelated. The aim is to estimate the spatially varying velocity, $C(x)$, by means of filter likelihood and the Metropolis-Hastings algorithm.

We follow the discretisation described in Section 5.1 for the advection equation.

The measurements' equation, (42), is discretised in time using the Euler-Maruyama scheme to yield

$$\delta y_{t_n} = H(t_n)u(x, t_n)\delta t + R^{1/2}(t_n)\delta\eta_{t_n}, \tag{43}$$

upon substituting $R = QQ^T/\delta x$ and where $\mathbb{E}[\delta\eta_{t_n}\delta\eta_{t_n}^T] = I_{r\times r}\delta t$. The observation likelihood pdf for the KBF is Gaussian since the initial condition and the observation errors are Gaussian. So is the posterior pdf. With observation increments expressed as in (43), the observation increment likelihood pdf is

$$\pi\left(\delta y_{t_0:t_N}\middle|\tilde{u}_{t_N}, \lambda_k\right) \propto \prod_{n=0}^{N}\exp\left(-\frac{1}{2}\left\|\delta y_{t_n} - H(t_n)\tilde{u}_{t_n}\delta t\right\|^2_{\delta t R(t_n)}\right). \tag{44}$$

Similarly, the filter forecast pdf is

$$\pi\left(u_{t_N}\middle|\delta y_{t_{N-1}:t_0}\right) \propto \prod_{n=0}^{N}\exp\left(-\frac{1}{2}\left\|u_{t_n} - \tilde{u}_{t_n}\right\|^2_{P_{t_n}}\right). \tag{45}$$

The next step is to implement a KBF and EnKBF and to obtain the likelihood at each time step. This is followed by implementing a Metropolis-Hastings algorithm.

The same is repeated but with EnKBF in the place of KBF. Algorithm 3 is the pseudocode showing the basic steps.

Results for the first 2 parameters are shown in Figure 3. The results in Figure 3 show that the EnKBF performs like the KBF filter. This agrees with the theory (see, for example, [17]) that the EnKBF yields optimal results in the limit $M \longrightarrow \infty$. It is also noteworthy that the Metropolis-Hastings algorithm converges to the true parameter estimate. This can be seen in Figure 3(a), for example, where

Require: δt, $\aleph_m$, M, N, $\{u_{t_0}^i\}_{i=1}^M$, π_{t_n}, and λ_0.
Ensure: $\{\lambda_k\}_{k=1}^T$.
1: **for** $k = 1$ **to** N **do**
2: Draw $\tilde{\lambda} \sim \mathcal{N}(\lambda_{i,k} \cos(\phi), (\omega/\aleph_m) \sin(\phi))$ $\forall i = 1, 2, \cdots, 2\aleph_m + 1$.
3: Compute $C_k(x) = \exp(g(x, \lambda_k))$.
4: **for** $n = 1$ **to** T, $\delta t > 0$ **do**
5: **for** $i = 1$ **to** M **do**
6: Run a single-step EnKBF prediction ensemble $\tilde{u}_{t_{n+1}}^i = u_{t_n}^i + F(t_n, \lambda_k) u_{t_n}^i \delta t$
7: Run a single-step EnKBF analysis ensemble $u_{t_{n+1}}^i = \tilde{u}_{t_{n+1}}^i + P_{t_n} H^T(t_n) R^{-1}(t_n)(dy_{t_n} + \varepsilon_i - H(t_n)\tilde{u}_{t_n}^i \delta t)$
8: **end for**
9: Compute prediction ensemble mean: $\bar{u}_{t_n} = 1/M \sum_{i=1}^M \tilde{u}_{t_n}^i$.
10: Compute analysis ensemble mean: $\hat{u}_{t_n} = 1/M \sum_{i=1}^M u_{t_n}^i$.
11: Compute covariance: $P_{t_n} = 1/M - 1 \sum_{i=1}^M (u_{t_n}^i - \hat{u}_{t_n})(u_{t_n}^i - \hat{u}_{t_n})^T$.
12: **end for**
13: Metropolis-Hastings
14: Compute $\alpha_{\text{ratio}} = (\pi(\delta y_{t_N:t_0} | \bar{u}_{t_N}, \lambda_k)/\pi(\delta y_{t_N:t_0} | \bar{u}_{t_N}, \lambda_{k-1}))(\rho_\lambda(\tilde{\lambda})/\rho_\lambda(\lambda_k))$
15: Compute $\alpha = \min(1, \alpha_{ratio})$
16: **if** $\alpha > U(0, 1)$ **then**
17: $\tilde{\lambda} = \lambda_k$
18: **else**
19: $\lambda_k = \tilde{\lambda}$
20: **end if**
21: **end for**

Algorithm 3: EnKBF likelihood with MH.

the filter estimate converges after about 100 parameter draws.

We now look at the errors in the parameter estimates, the better to see the performance of the filters for the 21 hyperparameters.

Figure 4(b) plots the boxplots showing the dispersion of parameter estimates resulting from the use of EnKBF and the root mean square errors for parameter estimates for both the EnKBF and KBF. The RMSE values are computed as follows.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{k=1}^{N} \left(\lambda_{i,k} - \lambda_i^{true}\right)^2}, \tag{46}$$

where $\lambda_{i,k}$ is the estimate of the ith parameter at the Metropolis-Hastings cycle k and λ_i^{true} is the true$i$th parameter.

The RMSE for both the KBF and EnKBF, as shown in Figure 4(b), indicate that the performance of EnKBF matches that of the KBF for the 21 parameters. These heuristic results corroborate the theoretical findings. The boxplot, Figure 4(a), shows the dispersion of parameter samples in the case when EnKBF is used. The result indicates that the estimates match the true parameter values, with not so many outliers. This is indicative of the performance of not only EnKBF but also the Metropolis-Hastings procedure in locating the true parameter values and ensuring that no large excursions are made from the true parameter values.

We now implement Algorithms 2 and 3 for the discretised wave equation, (33).

Example 2. Wave equation.

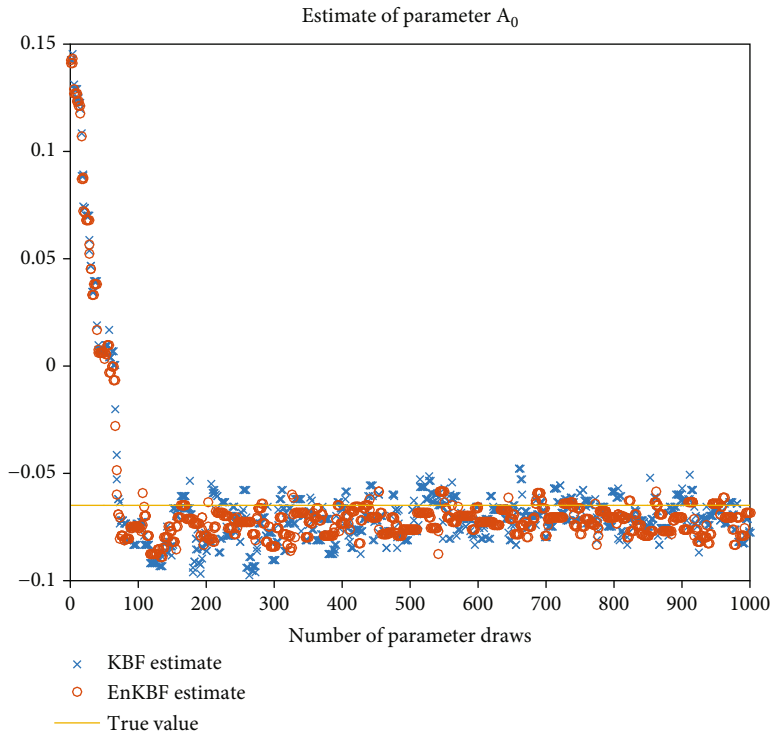
We take the wave equation of Section 5.2 and the associated initial conditions, Equations (30a) and (30b). The measurements are given by (42). The initial values of u_t, $\{\beta_t\}$, and $\{\eta_t\}$ are uncorrelated. The aim is to estimate the spatially varying velocity, $C(x)$, by means of filter likelihood and the Metropolis-Hastings algorithm.

The discretisation of the wave equation is as shown in Section 5.2. Figures 5(a) and 5(b) show the results.

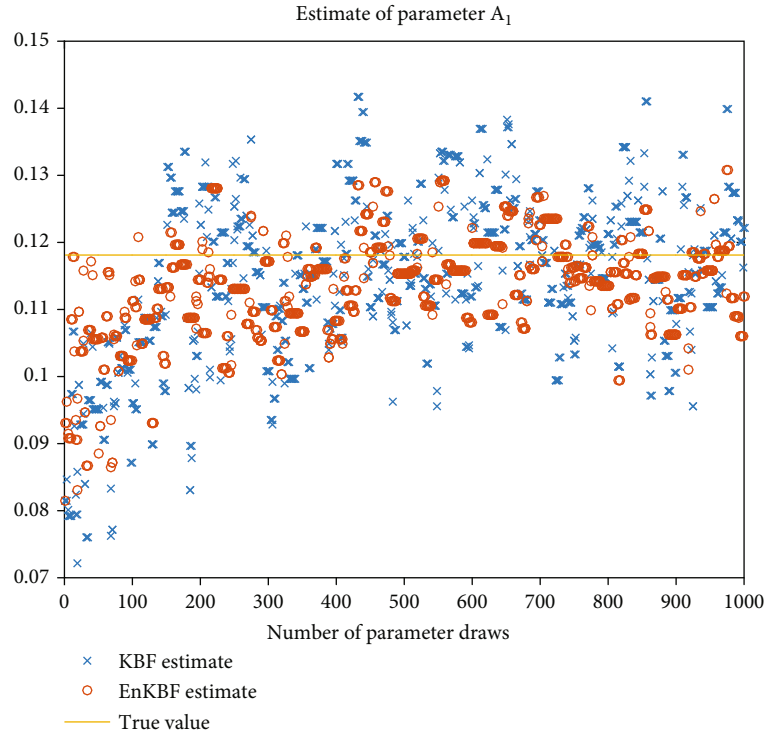
The results in Figure 5 indicate a close match in the performance of the EnKBF and KBF. This is as was anticipated in the theoretical findings, some of which are found in [17]. Notice also that the two filters converge to the true parameter values after a few parameter draws (about 50 in Figure 5(a) and 100 in Figure 5(b))—which is indicative of the robustness of the Metropolis-Hastings algorithm atop the EnKBF and KBF filters. The results also show that there are no wide excursions from the true parameter values, which testifies to the good performance of Algorithms 2 and 3.

In Figure 6 are plotted the boxplots showing the dispersion of the 21 hyperparameter estimates resulting from the use of EnKBF and the root mean square errors for parameter estimates for both the EnKBF and KBF.

The EnKBF elicits an optimal performance as can be seen in Figure 6(b) where the RMSEs for both the EnKBF



(a)



(b)

FIGURE 3: Plots for velocity parameters for the first three parameters in λ obtained using KBF and EnKBF and run for 1000 time steps and for 1000 Metropolis-Hastings cycles. The number of particles used for EnKBF is $M = 1000$, the time step size used in both filters is $dt = 0.01$, $\mu = 0.001$, and 100 grid points are used. The plots indicate that the estimates, for both filters, converge after about 100 iterations.

and the optimal filter (KBF) match for all the 21 hyperparameters. This also, as in the advection equation above, is in agreement with the theoretical findings that the EnKBF attains an optimal estimate in the limit $M \longrightarrow \infty$. The boxplot, Figure 6(a), shows that the mean of EnKBF parameter estimates matches the true parameter values. What is more,

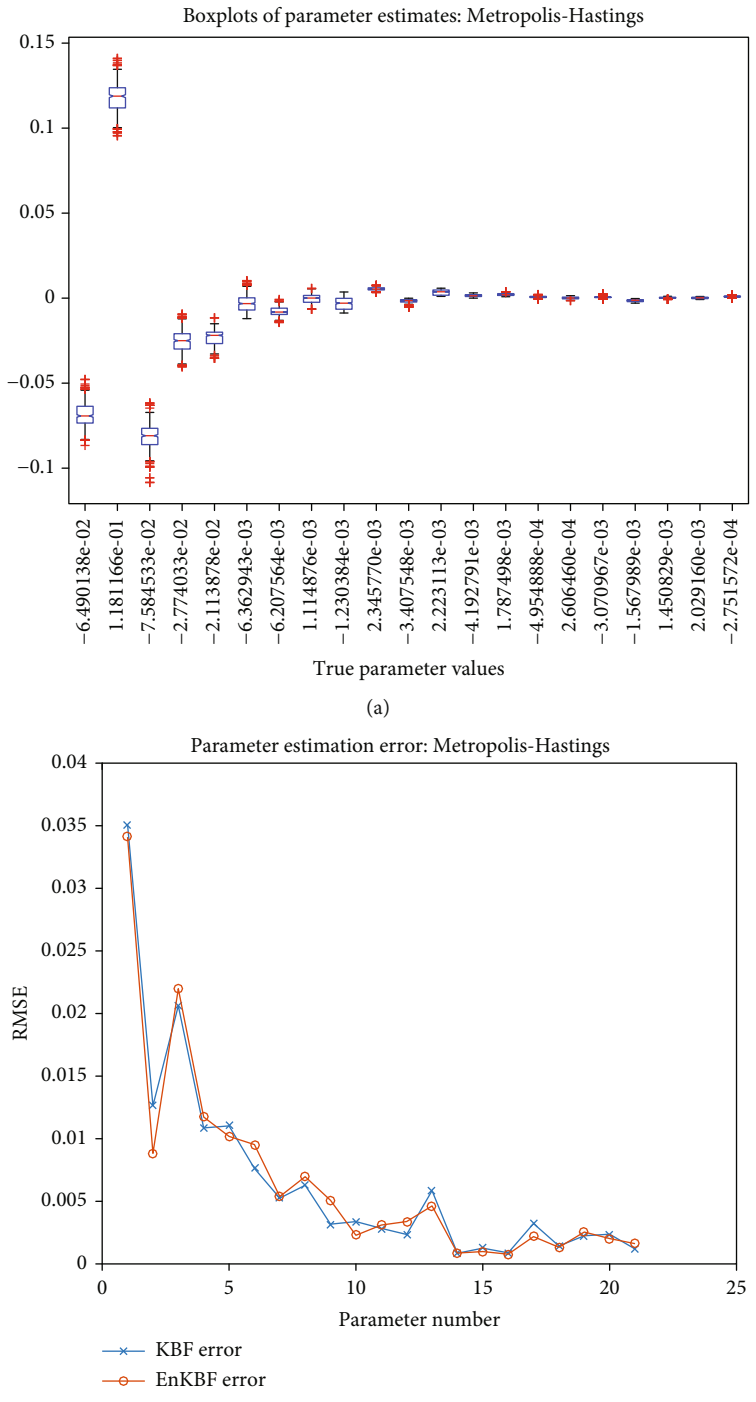


FIGURE 4: (a) Boxplot for the 21 velocity parameters for the EnKBF run for 1000 time steps with 1000 Metropolis-Hastings cycles. A burn-in of 500 parameter draws is discarded. The stochastic advection equation model is used with the following settings: $L = 2\pi$, 100 grid points, $\delta t = 0.01$, $M = 1000$ particles, $\mu = 0.001$, and localization radius of 10 grid points. (b) A plot of the root mean square error for the 21 hyperparameter estimates obtained using EnKBF and KBF. The plot indicates that the performance of EnKBF matches that of KBF in this setting.

there are not many outliers in the estimates. All these show that the EnKBF-Metropolis-Hastings algorithm is robust.

In the next section, we apply the concepts on dual estimation to stochastic hyperbolic PDEs, which, in this case, are the advection and the wave equations.

8. Simultaneous Estimation of the State and Spatially Varying Parameters

The dual filter (see Section 4.2 for details) can be adapted to allow for estimating both the state and spatially varying

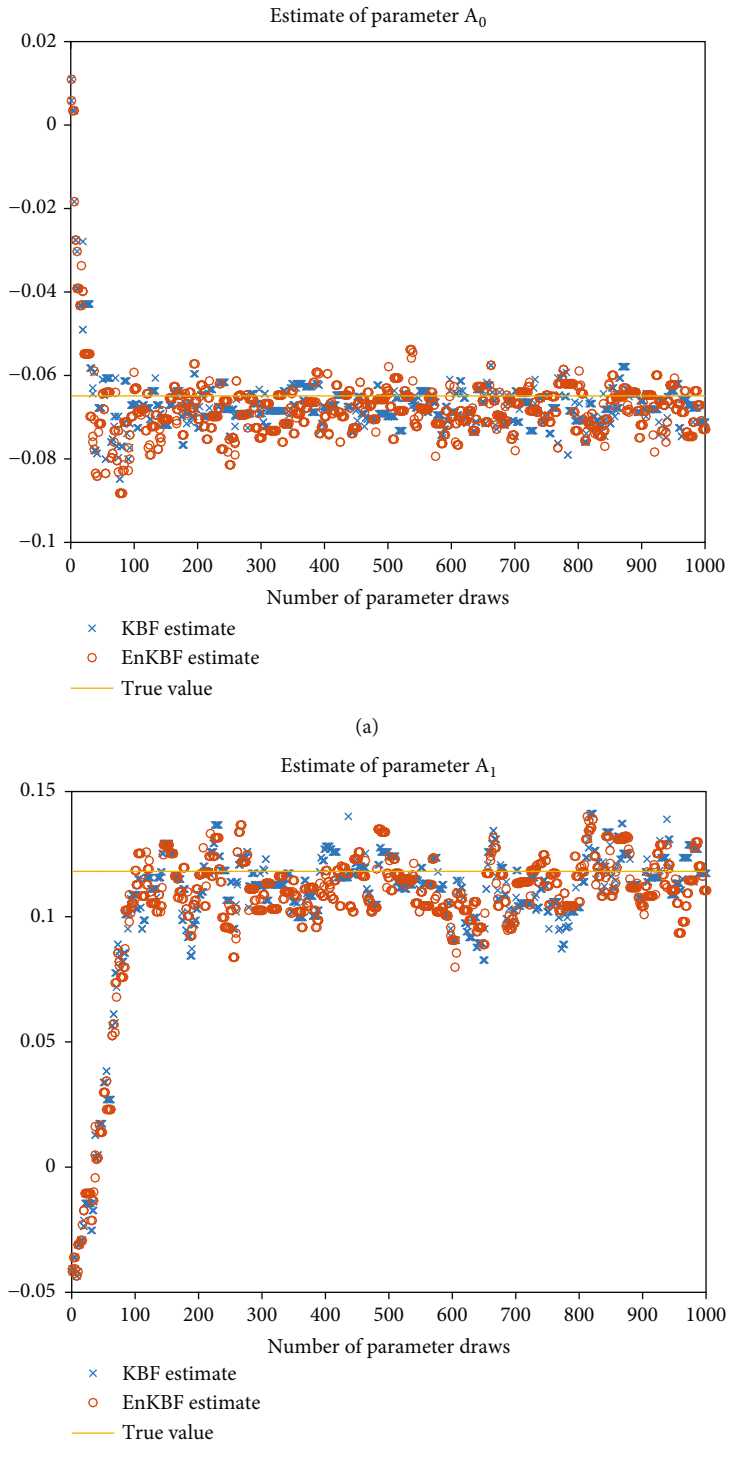


FIGURE 5: Plots for velocity parameters for the first three parameters in λ obtained using KBF and EnKBF and run for 1000 time steps and 1000 Metropolis-Hastings cycles. The number of particles used for EnKBF is $M = 1000$, the time step size used in both filters is $dt = 0.01$, $\mu = 0.001$, and 100 grid points are used. The plots indicate that the estimates, for both filters, converge after about 100 iterations.

parameters, contemporaneously. The idea here is to replace the parameters in the dual filter with hyperparameters of the varying parameters to be approximated. The hyperparameters are then updated simultaneously and in parallel, with the state at each iteration, where one filter estimates the state and the other filter updates the hyperparameters—with each filter making use of the outcome of the other. To illustrate this argument, we turn

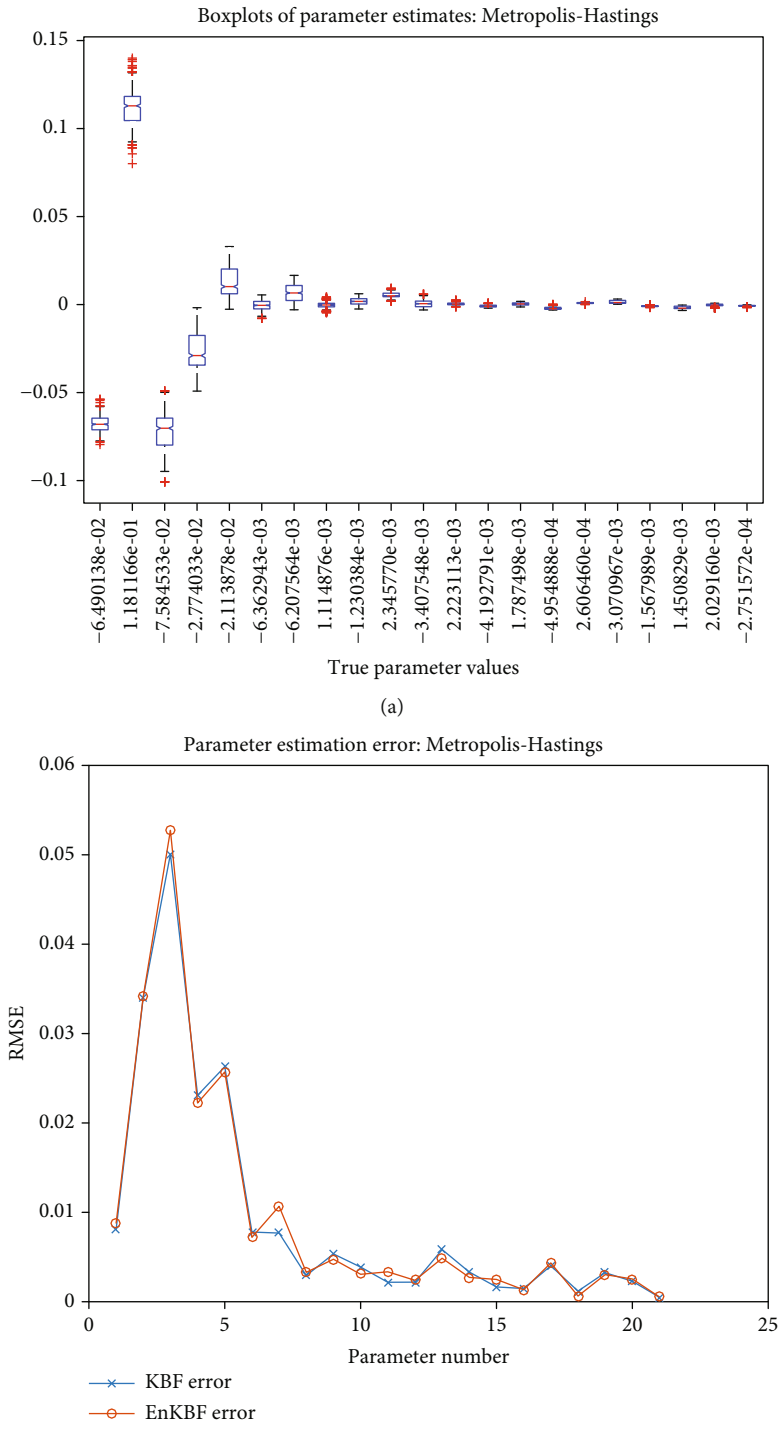


FIGURE 6: (a) Boxplot for velocity parameters for the EnKBF run for 1000 time steps and 1000 Metropolis-Hastings cycles. A burn-in of 500 parameter draws is discarded. The stochastic wave equation model is used with the following settings: $L = 2\pi$, 100 grid points, $\delta t = 0.01$, $M = 1000$ particles, $\mu = 0.001$, and localization radius of 10 grid points. (b) A plot of the root mean square error for the parameter estimates obtained using EnKBF and KBF. The plot indicates that the performance of EnKBF matches that of KBF in this setting.

to the advection and wave equation described in Examples 1 and 2, respectively, and use the KBF-EnKBF dual filter—in which the state is propagated and updated by means of the KBF whilst the hyperparameters are updated using EnKBF—and ENKBF dual filter—where both the state and the hyperparameters are propagated and updated using two EnKBFs running in parallel. The spatially varying velocity is as shown in Section 6.

Require: $u_{t_0}^j$, $\lambda_{t_0}^j$, $w_{t_0}^j = (1/L) \quad \forall j \in \{1, 2, \cdots, L\}$, P_{t_0}, and $\delta y_{[t_0,t_T]}$.
Ensure: $\widehat{u}_{[t_0,t_N]}$, $\widehat{\lambda}_{[t_0,t_N]}$.
1: **for** $n = 1$ **to** N, $\delta t > 0$ **do**
2: **for** $j = 1$ **to** L **do**
3: Update $\widehat{u}_{t_n}^j$ using Equations (47a) and (47b)
4: Update parameters $\lambda_{t_n}^j$ using (48)
5: **end for**
6: Compute $\widehat{\lambda}_{t_n} = 1/L\sum_{j=1}^{L}\lambda_{t_n}^j$
7: Compute $\widehat{u}_{t_n} = 1/L\sum_{i=1}^{L}\widehat{u}_{t_n}^j$
8: **end for**

Algorithm 4: KBF-EnKBF dual filter.

Require: $u_{t_0}^{i,j}$, $\lambda_{t_0}^j$, $w_{t_0}^j = (1/L) \quad \forall i \in \{1, 2, \cdots, M\}\ j \in \{1, 2, \cdots, L\}$, P_{t_0}, and $\delta y_{[t_0,t_T]}$.
Ensure: $\widehat{u}_{[t_0,t_N]}$, $\widehat{\lambda}_{[t_0,t_N]}$.
1: **for** $n = 1$ **to** N, $\delta t > 0$ **do**
2: **for** $j = 1$ **to** L **do**
3: **for** $i = 1$ **to** M **do**
4: Calculate $u_{t_n}^{i,j}$ using (50)
5: **end for**
6: Update $\widehat{u}_{t_n}^j$ using Equation (49c)
7: Update parameters $\lambda_{t_n}^j$ using (48)
8: **end for**
9: Compute $\widehat{\lambda}_{t_n} = 1/L\sum_{j=1}^{L}\lambda_{t_n}^j$
10: Compute $\widehat{u}_{t_n} = 1/L\sum_{i=1}^{L}\widehat{u}_{t_n}^j$
11: **end for**

Algorithm 5: EnKBF dual filter.

In the KBF-EnKBF dual filter, we update, for every jth parameter particle $\lambda_t^j \in \{\lambda_t^j\}_{j=1}^L$, the state estimate, $\widehat{u}_t^j$, using the KBF; that is,

$$d\widehat{u}_t^j = F(t)\widehat{u}_t^j dt + P_t H^T(t)R^{-1}(t)\left(dy_t - H(t)\widehat{u}_t^j dt\right), \quad (47a)$$

$$\begin{aligned} dP_t = {} & F(t)P_t dt + P_t F^T(t)dt + G(t)G^T(t)dt \\ & - P_t H^T(t)R^{-1}(t)H(t)P_t dt. \end{aligned} \quad (47b)$$

The parameters are updated using the EnKBF; that is, each parameter hypothesis, λ_t^j, is updated using

$$d\lambda_t^j = D_t^L H(t)R^{-1}(t)\left(dy_t - 0.5\left(H(t)\widehat{x}_t^j + H(t)\widehat{x}_t\right)dt\right); t_0 \le t, \quad (48)$$

where

$$\widehat{\lambda}_t = \frac{1}{L}\sum_{j=1}^{L}\lambda_t^j; t_0 \le t, \quad (49a)$$

$$D_t^L = \frac{1}{L-1}\sum_{i=1}^{L}\left(\lambda_t^j - \widehat{\lambda}_t\right)\left(\widehat{u}_t^j - \widehat{u}_t\right)^T; t_0 \le t, \quad (49b)$$

where

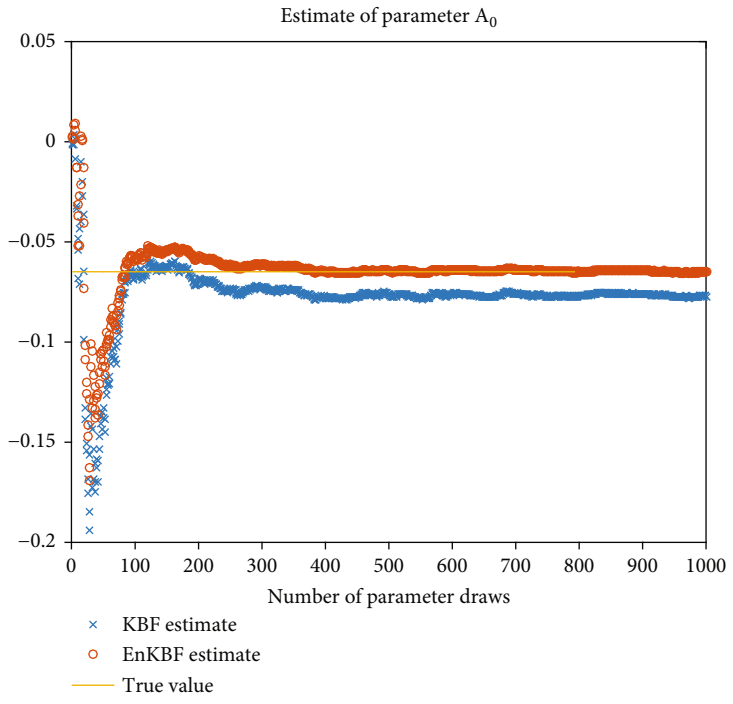
$$\widehat{u}_t^j = \frac{1}{M}\sum_{i=1}^{M}u_t^{i,j}, \quad (49c)$$

$$\widehat{u}_t = \frac{1}{L}\sum_{i=1}^{L}\widehat{u}_t^j. \quad (49d)$$

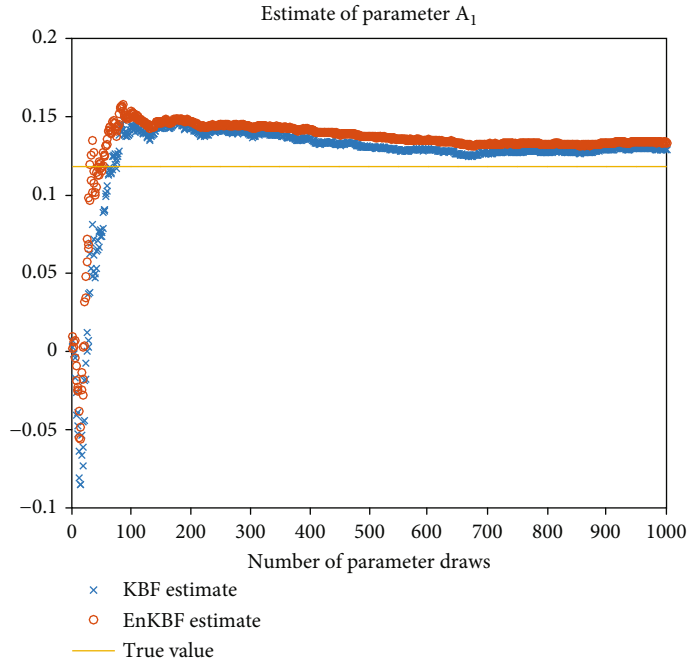
Algorithm 4 gives a summary of the KBF-EnKBF dual filter.

The EnKBF dual filter consists of an update of M particles of the state, $u_t^{i,j} \in \{u_t^{i,j}\}_{i,j=1}^{M,L}$, for every parameter particle, $\lambda_t^j \in \{\lambda_t^j\}_{j=1}^L$, using the EnKBF; that is,

$$\begin{aligned} du_t^{i,j} = {} & F(t)u_t^{i,j}dt + G(t)d\beta_t^{i,j} \\ & + P_t^M H^T(t)R^{-1}(t)\left(dy_t + R^{1/2}(t)\eta_t^{i,j} - H(t)u_t^{i,j}dt\right), \end{aligned} \quad (50)$$



(a)



(b)

FIGURE 7: (a, b) are plots for velocity parameters for the first two parameters in λ obtained using KBF-EnKBF and EnKBF dual filters, applied to the advection equation, both run for 1000 time steps. The number of particles, for the state and hyperparameters, used in EnKBF is $M = 1000$ and $L = 1000$; the time step size used in both filters is $dt = 0.01$. $\mu = 0.001$ and 100 grid points are used. Compared to the results obtained using the Metropolis-Hastings algorithm (Section 7), the dual filters register a better performance, at least in this example.

where $\{\eta_t^{i,j}, t_0 \leq t\}$ and $\{\beta_t^{i,j}, t_0 \leq t\}$ are standard Brownian motion vector processes. The parameters are updated using the EnKBF given by (48). The summary of the EnKBF dual filter is given in Algorithm 5.

Figures 7(a) and 7(b) show the results for the first two parameters in (39) when the dual filters are applied to the advection equation.

Evidently, from Figure 7, the performance of the EnKBF and KBF-EnKBF dual filters almost matches. Both filters

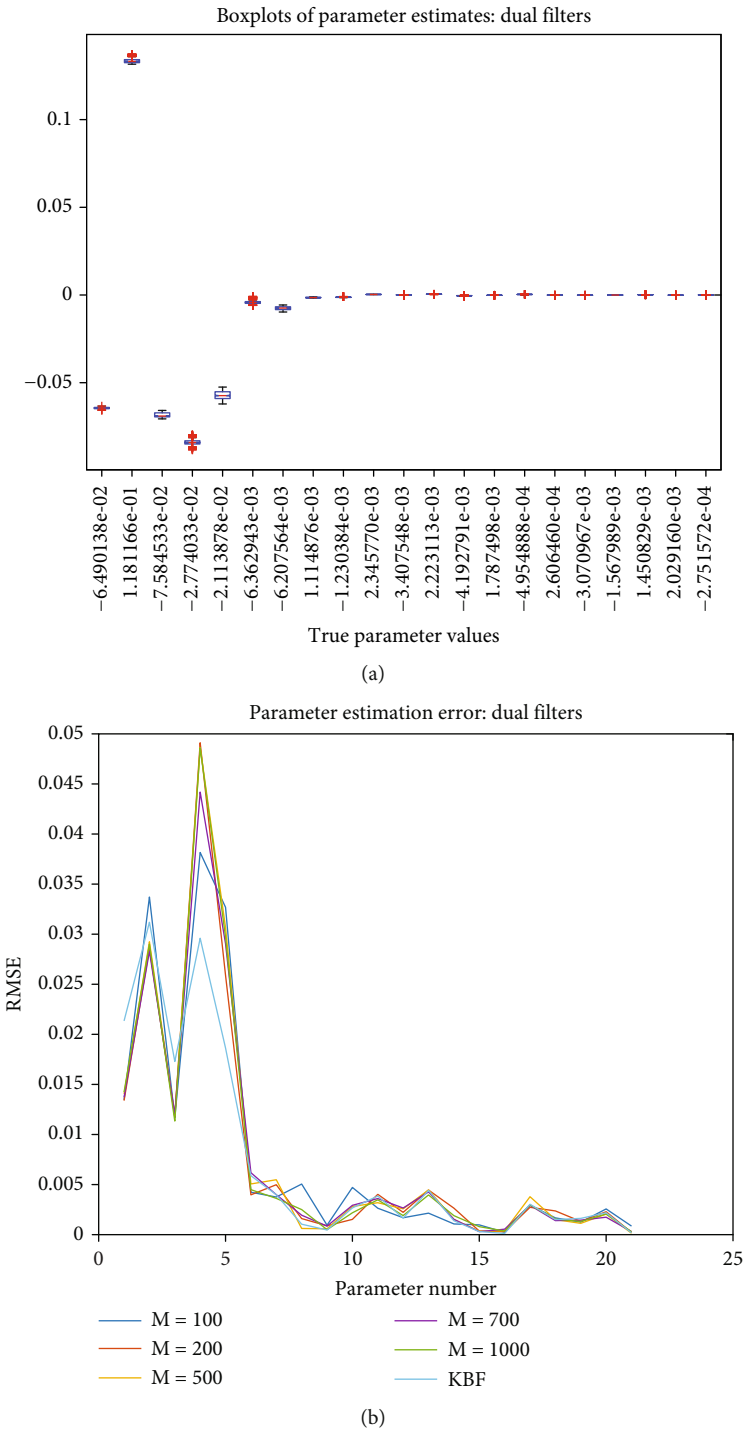
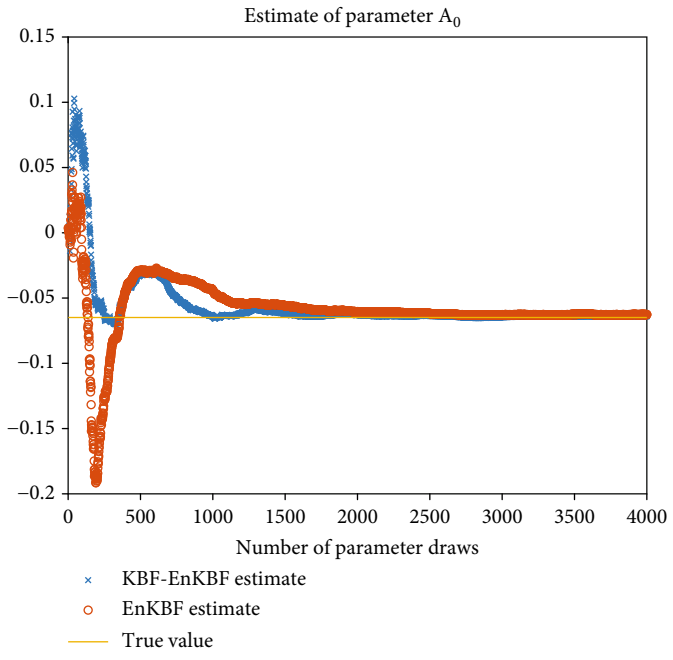


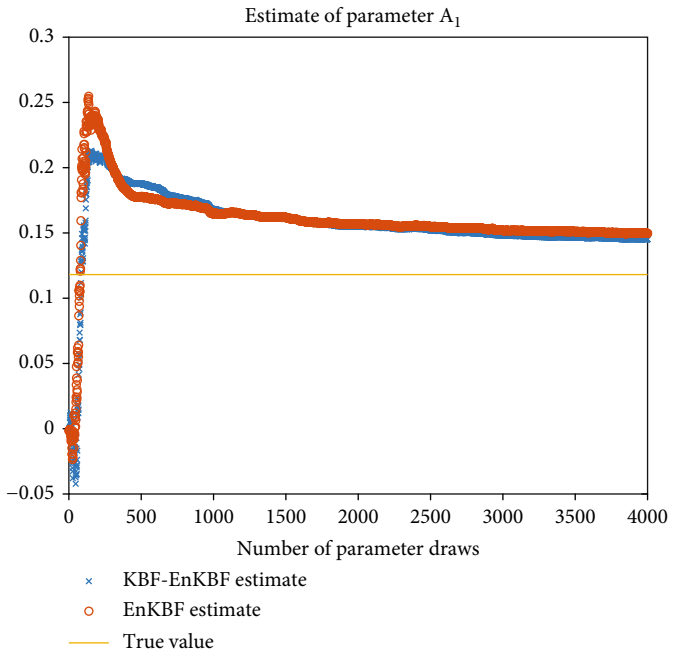
FIGURE 8: (a, b) are, respectively, the boxplots showing the distribution of hyperparameter estimates of the EnKBF dual filter, applied to the advection equation, after a burn-in of 500 iterations and the plots of the root mean square error for the hyperparameter estimates obtained using KBF-EnKBF and EnKBF (with different ensemble sizes) dual filters.

converge to the true estimate after a few iterations (about 100 in Figure 7(a)). This is another testament to the fact that EnKBF attains optimal estimates at high ensemble values. Moreover, both the EnKBF and KBF-EnKBF parameter estimates are not much spread as compared to the previous case where Metropolis-Hastings was used. This is more evident in the following results for the 21 hyperparameters estimated.

In the following panels are plotted the root mean square errors for parameter estimates for both the EnKBF and KBF-EnKBF dual filters and the boxplots showing the dispersion of parameter estimates resulting from the



(a)



(b)

FIGURE 9: (a, b) are plots for velocity parameters for the first two parameters in λ obtained using KBF-EnKBF and EnKBF dual filters, applied to the wave equation (Example 2), with both filters run for 1000 time steps. The number of particles, for the state and hyperparameters, used in the EnKBF filter is $M = 1000$; the time step size used in both filters is $dt = 0.01$. $\mu = 0.001$ and 100 grid points are used. Compared to the results obtained using the Metropolis-Hastings algorithm (Section 7), the dual filters register a dismal performance, at least in this example.

use of the EnKBF dual filter applied to the advection equation.

That the boxplots of EnKBF dual filter parameter estimates have short whiskers (see Figure 8(a)) and few outliers and that the estimates match the true parameter values indicate that the EnKBF dual filter is robust in this setting. The EnKBF dual filter registers a slight variation in RMSE from the KBF-EnKBF dual filter. This indicates that the EnKBF, in this setting, performs optimally.

We now repeat the same procedure but with the wave equation described in Section 5.2 in place of the advection equation. The following panels show the results. The results

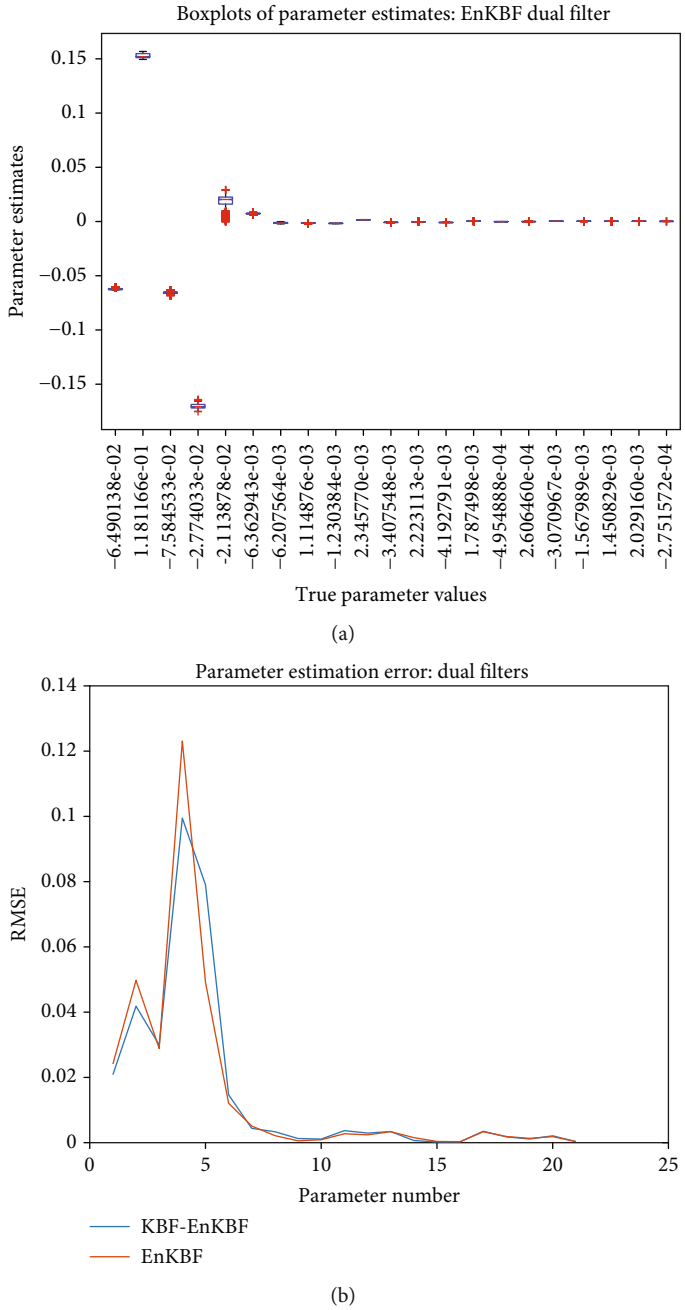


Figure 10: (a, b) are, respectively, the boxplots showing the distribution of the 21 hyperparameter estimates of the EnKBF dual filter after a burn-in of 500 iterations and plots of the root mean square error for the hyperparameter estimates obtained using KBF-EnKBF and EnKBF (with different ensemble sizes) dual filters, applied to the wave equation (Example 2).

shown in Figure 9 are indicative of a dismal performance of the two dual filters—KBF-EnKBF and EnKBF dual filters—when applied to the wave equation as compared to the results obtained when the dual filters are applied to the advection equation (see Figure 7). We note that the wave equation is partially observed; that is, only u is observed in the discretised wave equation, (33), whereas the advection equation is fully observed. Furthermore, the number of unknowns in the state-parameter system of the wave equation is 300 whilst that in the advection equation is 200. These account for the dismal performance of the KBF-EnKBF and EnKBF dual filters when applied to the wave equation.

In Figure 10 are plotted the root mean square errors for parameter estimates for both the EnKBF and KBF and the boxplots showing the dispersion of parameter estimates resulting from the use of EnKBF.

9. Conclusions

We have considered a special case of parameter estimation where the parameter to be estimated is spatially varying.

Such a case arises, for example, in the velocity of a wave travelling through an inhomogeneous media. We proposed and studied two approaches: the use of filter likelihood and the Metropolis-Hastings procedure and joint estimation of state and parameters. The parameter is expressed as a Fourier series with constant coefficients. The coefficients are approximated and then substituted back to the Fourier series to obtain an approximation of the velocity. The Kalman-Bucy filter and the ensemble Kalman-Bucy filter are used. The filter likelihood with the Metropolis-Hastings procedure registers a better performance compared to the joint estimation procedure in both advection and wave equations. From the foregoing, Metropolis-Hastings with the filter evidence performs well in estimation of parameters compared to the dual filters—especially when the number of unknowns is large. This is indicative of the robustness of the Metropolis-Hastings algorithm in searching, and remaining in, the high probability region of the state-space.

Data Availability

The data is obtained from computer simulation using MATLAB software. The codes generating the results can be presented on request put to the author via dangwenyi@mmust.ac.ke.

Conflicts of Interest

The author declares that they have no conflicts of interest.

Acknowledgments

The author would like to acknowledge the assistance of Professor Dr. Sebastian Reich, of Potsdam University, Germany, in supervising their doctoral research [17], of which this paper is part. The manuscript of this work can be found in [18].

References

[1] G. A. Einicke, *Smoothing, Filtering and Prediction: Estimating the Past, Present and Future*, BoD–Books on Demand, Rijeka, Croatia, 2012.

[2] S. Särkkä, *Bayesian Filtering and Smoothing*, Cambridge University Press, Cambridge, 2014.

[3] J. Lewis, S. Lakshmivarahan, and S. Dhall, *Dynamic Data Assimilation: a Least Squares Approach*, Cambridge University Press, 2009.

[4] Z. Zhang, "Parameter estimation techniques: a tutorial with applications to conic fitting," *Image and Vision Computing*, vol. 15, no. 1, pp. 59–76, 1997.

[5] C. Robert and G. Casella, *Monte Carlo Statistical mMethods*, Springer-Verlag, New York, 2004.

[6] J. W. C. V. Lint, S. P. Hoogendoorn, and A. Hagyi, "Dual EKF state and parameter estimation in multi-class first-order traffic flow models," in *Proceedings of the17th World Congress', The International Federation of Automatic Control*, Seoul, Korea, 2008.

[7] H. Lü, Z. Yu, Y. Zhu, S. Drake, Z. Hao, and E. A. Sudicky, "Dual state-parameter estimation of root zone soil moisture by optimal parameter estimation and extended Kalman filter data assimilation," *Advances in Water Resources*, vol. 34, no. 3, pp. 395–406, 2011.

[8] H. Moradkhani, S. Sorooshian, H. V. Gupta, and P. R. Houser, "Dual state-parameter estimation of hydrological models using ensemble Kalman filter," *Advances in Water Resources*, vol. 28, no. 2, pp. 135–147, 2005.

[9] E. Wan and T. Nelson, "Dual Kalman filtering methods for nonlinear prediction, smoothing and estimation," *Advances in Neural Information Processing Systems*, vol. 9, 1997.

[10] D. Angwenyi, J. de Wiljes, and S. Reich, "Interacting particle filters for simultaneous state and parameter estimation," 2017, https://arxiv.org/abs/1709.09199.

[11] M. Stuart, *Finite difference approximation for linear stochastic partial differential equations with method of lines*, Munich Personal RePEc Archive, Paper No. 3983, 2007.

[12] E. Allen, S. Novosel, and Z. Zhang, "Finite element and difference approximation of some linear stochastic partial differential equations," *Stochastics: An International Journal of Probability and Stochastic Processes*, vol. 64, no. 1-2, pp. 117–142, 1998.

[13] R. Courant, E. Isaacson, and M. Rees, "On the solution of nonlinear hyperbolic differential equations by finite differences," *Communications on Pure and Applied Mathematics*, vol. 5, pp. 243–255, 1952.

[14] E. Hairer, C. Lubich, and G. Wanner, "Geometric numerical integration illustrated by the Störmer–Verlet method," *Acta Numerica*, vol. 12, pp. 399–450, 2003.

[15] S. Reich, "Backward error analysis for numerical integrators," *SIAM Journal of Numerical Analysis*, vol. 36, no. 5, pp. 1549–1570, 1999.

[16] J. Chen and H. Wu, "Estimating of time-varying parameters in deterministic dynamic models," *Statistica Sinica*, vol. 18, pp. 987–1006, 2008.

[17] D. Angwenyi, *Time-continuous state and parameter estimation with application to hyperbolic SPDEs*, doctoral thesis, Universität Potsdam, 2019.

[18] D. Angwenyi, "Estimation of spatially varying parameters with application to hyperbolic SPDEs," 2021, https://arxiv.org/abs/2107.07246.