

CONCURRENT MULTIPARAMETER LEARNING
DEMONSTRATED ON THE KURAMOTO–SIVASHINSKY
EQUATION*BENJAMIN PACHEV[†], JARED P. WHITEHEAD[‡], AND SHANE A. MCQUARRIE[†]

Abstract. We develop an algorithm based on the nudging data assimilation scheme for the concurrent (on-the-fly) estimation of scalar parameters for a system of evolutionary dissipative partial differential equations in which the state is partially observed. The algorithm takes advantage of the error that results from nudging a system with incorrect parameters with data from the true system. The intuitive nature of the algorithm makes its extension to several different systems immediate, and it allows for recovery of multiple parameters simultaneously. We test the method on the Kuramoto–Sivashinsky equation in one dimension and demonstrate its efficacy in this context.

Key words. data assimilation, parameter recovery, Kuramoto–Sivashinsky equation, nudging

MSC codes. 34D06, 35Q30, 35Q35, 37C50

DOI. 10.1137/21M1426109

1. Introduction. Many mathematical models have parameters that must be identified correctly to match reality. This is often a difficult task, as such parameters are difficult to measure and/or noise in the measurement process precludes identifying the parameters exactly. Motivated by [8], we develop a robust algorithm that estimates multiple scalar parameters in a partial differential equation (PDE) when those parameters can be written as coefficients of the various terms in the PDE. As the current algorithm relies on the feedback control data assimilation mechanism introduced in [1], we simultaneously identify not only the parameters of the system but also obtain the dynamical state of the system based on reduced observations of the true state.

The method presented herein differs in important ways from other data-driven parameter recovery or equation discovery methods in the recent literature, the majority of which are *offline* procedures in the sense that they seek to learn models from a previously collected set of training data. The sparse identification of nonlinear dynamical systems (SINDy) framework [6] generates a library of nonlinear candidate functions and leverages sparsity-promoting optimization techniques to generate the model that, as a linear combination of only a few of the candidate functions, best explains a set of observed dynamics. The methodology has also been extended to a parametric setting through periodic sparse updates to existing models [38] or separating the parametric dependence from the state via group sparsity [40]. Recent advances in deep learning propose neural network models that build the residual of a known governing equation into the training loss function, informing the optimization problem of the desired physics [39]. Other deep learning approaches aim to directly approximate parameter-to-solution operators with neural networks [32, 34], establishing a mapping between

*Submitted to the journal's Methods and Algorithms for Scientific Computing section June 10, 2021; accepted for publication (in revised form) July 14, 2022; published electronically September 15, 2022.

<https://doi.org/10.1137/21M1426109>

Funding: The work of the second author was partially supported by the Simons Foundation travel grant 586788.

[†]Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX 78712 USA (benjaminpachev@gmail.com, shanemcq@utexas.edu).

[‡]Department of Mathematics, Brigham Young University, Provo, UT 84602 USA (whitehead@mathematics.byu.edu).

system parameters and solutions of the underlying PDEs. Such approaches massively overparameterize the system dynamics, endowing the resulting network with expressivity but often at the expense of training efficiency. Bayesian statistics yields yet another framework wherein a posterior distribution with associated uncertainties can be constructed for the parameters of the PDE, or even the solution itself (see [5], for example). The use of Markov chain Monte Carlo (MCMC) methods in this context provide robust methods for estimating the posterior distribution, and the advantages of the Bayesian approach are multifaceted [13]; however, such analyses are often computationally intensive as even the most advanced MCMC methods require thousands of evaluations of a numerical approximation to the PDE.

In contrast, the method we propose is a *concurrent* (on-the-fly) procedure, providing updates to the unknown parameters of a target model simultaneous to the assimilation of measurements. We consider the setting in which the form of the governing equations is known, and therefore have no need to construct a library of candidate terms or overparametrize the solution representation, and later extend the approach to a setting where less is known about the governing equation. We leverage well-established methods originally developed to determine the true state of a system when the model is fully known, but in our setting we additionally infer the unknown system parameters.

Our derivation is motivated by the Azouani–Olson–Titi (AOT) [1] algorithm developed for the continuous assimilation of limited data into a known dissipative dynamical system. Since [1], this approach has been adapted to numerous other systems (see [2, 4, 26], for example) and settings in which the observed variables are various subsets of the full system’s variables (e.g., [18, 19, 20, 21]) or where different types of observations and nudging are utilized (see [3, 31], for example). All of these extensions have broadened the applicability of the AOT algorithm but rely on the accurate representation of the underlying model parameters. Recently, data assimilation for an imperfect model has been considered in different contexts [8, 14, 17]. It is rigorously established for two distinct settings that when a parameter of the system is unknown, the error in the assimilation will still converge up to the error in the unknown parameter (see [8, 17]). The work [9] takes these concepts a step further, deriving analogous update formulae for the three parameters of the Lorenz equations [33] and carrying out a thorough numerical investigation to explore the limitations of these formulae. In addition, asymptotic convergence to the true parameter values under suitable conditions is rigorously guaranteed, giving the first instance where a rigorous mathematical framework is available for this approach to learning parameters. We expand on all of this previous work, developing an algorithm that works not only for a single unknown parameter, but for multiple unknowns, and which theoretically applies to any dissipative dynamical system.

Our algorithm is derived in a general setting and is applicable to a wide class of dissipative PDEs. In this work, we examine the efficacy of the algorithm numerically for the Kuramoto–Sivashinsky equation (KSE). KSE is selected because it presents chaotic spatial-temporal dynamics at a relatively low computational cost (see [7, 12, 15, 43] for example studies of the complicated dynamics that arise in KSE). In addition, KSE naturally provides the context for introducing several artificial parameters whose estimation is of significant import.

KSE has been derived in a variety of contexts, typically as a model for physical systems that are far from equilibrium. The original derivation of KSE was to study instabilities in a reaction diffusion system [29] and simultaneously on the instability and turbulent behavior of a single flame [41]. KSE also appears in the study of

plasmas [10, 30], in irregular flow down an inclined plane [42], and it is shown in [35] that KSE provides a general description for part of the dynamical evolution of systems following a certain type of bifurcation point. Mathematically, KSE is often used as an ideal setting to investigate nonlinear dynamics, coherent structures, long-time behavior of solutions, and the testing of methods developed to investigate these properties (see [11, 22, 23, 24, 25, 36], for example). Such earlier works have demonstrated that although KSE is simple and computationally cheap to simulate, it displays a wide variety of interesting dynamics similar to more complicated systems and is a reasonable testbed for the parameter estimation algorithm proposed below.

The paper is organized as follows. The derivation of the parameter learning algorithm is presented in section 2, and section 3 presents numerical simulations applied to KSE that demonstrate the apparent numerical properties of the algorithm. Section 4 draws conclusions and discusses extensions of these results to other systems and the potential for rigorous justification of the main algorithm.

2. Derivation of the algorithm. In this section, we develop (but do not prove convergence of) a parameter recovery algorithm in a general setting. Let $\Omega \subset \mathbb{R}^d$, $d \in \{1, 2, 3\}$, be an open set with Lipschitz continuous boundary $\partial\Omega$, and let $\mathcal{V} = L^2(\Omega)$ with symmetric inner product $\langle \cdot, \cdot \rangle$. All references to a vector space or properties of a vector space (orthogonality, etc.) referred to below will apply to $L^2(\Omega)$ unless specified otherwise.

2.1. Problem statement. Consider a PDE of the form

$$(2.1a) \quad \mathbf{u}_t + F(\mathbf{u}) + \sum_{k=1}^n \lambda_k G_k(\mathbf{u}) = \mathbf{f},$$

where $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is an unknown physical quantity (e.g., velocity), $\mathbf{u}_t = \frac{\partial}{\partial t} \mathbf{u}(\mathbf{x}, t)$ is its partial derivative in time, $\mathbf{f} = \mathbf{f}(\mathbf{x}, t)$ is a known forcing function, F and G_1, \dots, G_n are (potentially nonlinear) spatial differential operators, and $\lambda_1, \dots, \lambda_n \in \mathbb{R}$ are fixed real numbers. Throughout the derivation we assume that solutions of (2.1a) are classical [16]; i.e., we will not concern ourselves with establishing the regularity of the solutions.

Our goal is to use partial observations of the state \mathbf{u} to recover not only the full state but also the unknown parameters $\lambda_1, \dots, \lambda_n$. Before proceeding, we note that such a problem is not always well posed. For example, consider the following simplified, linear version of (2.1a),

$$\mathbf{u}_t - \lambda_1 \mathbf{u} + \lambda_2 A \mathbf{u} = \mathbf{f},$$

where A is a linear differential operator. If the true solution \mathbf{u}_δ is a time-independent eigenfunction of A with eigenvalue δ , then the inverse problem reduces to attempting to solve

$$(\lambda_2 \delta - \lambda_1) \mathbf{u}_\delta = \mathbf{f}$$

for λ_1 and λ_2 . This is a degenerate situation wherein λ_1 and λ_2 cannot be uniquely identified by any algorithm. This issue can occur in other settings (e.g., with time-dependent solutions and/or nonlinear terms); however, the problem that we examine in section 3 does not suffer from lack of identifiability.

Motivated by [8, 17] and in tandem with (2.1a), we consider the coupled system

$$(2.1b) \quad \mathbf{v}_t + F(\mathbf{v}) + \sum_{k=1}^n \hat{\lambda}_k G_k(\mathbf{v}) = \mathbf{f} + \mu(I_h(\mathbf{u}) - I_h(\mathbf{v})),$$

where $\hat{\lambda}_k = \hat{\lambda}_k(t)$ is the estimate for λ_k (which may evolve in time), I_h is a linear interpolation operator corresponding to partial observations of the true system state, and $\mu > 0$ is called the nudging parameter. The subscript h refers to the level of resolution of the observation operator; i.e., $h \rightarrow 0$ would indicate observation of all relevant scales. This is exactly the AOT formulation of [1], except that $\hat{\lambda}_k(t) \neq \lambda_k$, and these parameter estimates are updated in time along with the state estimate for \mathbf{v} . Our goal is to have $\mathbf{v}(t) \rightarrow \mathbf{u}(t)$ and $\hat{\lambda}_k(t) \rightarrow \lambda_k$ up to an acceptable level of precision as t increases. Following arguments similar to those put forward in [8, 17], under certain practical assumptions on (2.1a) the convergence of $\mathbf{v}(t) \rightarrow \mathbf{u}(t) + \mathcal{O}(|\lambda_k - \hat{\lambda}_k(t)|)$ is guaranteed; however, obtaining the desired parameter convergence $\hat{\lambda}_k(t) \rightarrow \lambda_k$ requires an update formula for the parameters. This is the main subject of this work.

In [8], two different update formulae are proposed for the 2D Navier–Stokes equations when the single unknown parameter λ is the viscosity of the system. Both approaches assume that $\mathbf{w} := \mathbf{u} - \mathbf{v}$ is sufficiently small so that quadratic terms in the error can be neglected, leading to an update formula for $\hat{\lambda}$. Such an assumption is valid, assuming that $\hat{\lambda}$ is initially chosen close enough to λ so that \mathbf{w} is indeed small. Critically, this update formula is applicable only when $n = 1$, i.e., there is a single parameter to update. The approach we take here does not suffer from either of these restrictions: we make no assumption on the magnitude of \mathbf{w} , and we can simultaneously recover multiple parameters at once.

2.2. Parameter estimation to guide synchronization. We assume that we can only observe $I_h(\mathbf{u})$ rather than the full state \mathbf{u} . Furthermore, the time derivative $I_h(\mathbf{u}_t)$ is not directly available (i.e., we do not have direct observations of \mathbf{u}_t) and must instead be numerically estimated via, e.g., finite differences of $I_h(\mathbf{u})$, which is feasible if we assume that I_h is a linear operator (this assumption is carried throughout the following derivation). Therefore, the metric of convergence that we use is the observable error, defined as $I_h(\mathbf{w}) = I_h(\mathbf{u}) - I_h(\mathbf{v})$. While convergence of the observable error to zero is not sufficient to imply convergence of the parameters or full state estimates, it is necessary. In other words, $\hat{\lambda}_k(t) \rightarrow \lambda_k$ and $\mathbf{w}(t) \rightarrow \mathbf{0}$ can only occur simultaneously when $I_h(\mathbf{w}(t)) \rightarrow \mathbf{0}$. To motivate the algorithm described here, we therefore examine the dynamics of $I_h(\mathbf{w})$.

Remark 2.1. We do not rigorously show that $I_h(\mathbf{w}(t)) \rightarrow 0$ implies that $\mathbf{w}(t) \rightarrow 0$. However, traditional Foias–Prodi-type estimates typically used to guarantee convergence of AOT-type algorithms apply in this setting with a correction term due to the differences in the true parameters and the estimated parameters. Correction terms are considered in a different context in both [8] and [17], although the current setting is more general. The primary difficulty is that while such a correction term can certainly be estimated, it is updating in time; i.e., convergence of $\mathbf{w}(t) \rightarrow 0$ is dependent on the convergence of $|\lambda_k - \hat{\lambda}_k(t)| \rightarrow 0$ and vice versa. That is, the dual convergence of the state and the parameters is not easily decoupled. This makes obtaining the usual error estimates challenging.

Combining (2.1a) and (2.1b), the evolution of the observable error can be written as

$$\begin{aligned} \frac{d}{dt} I_h(\mathbf{w}) &= I_h(\mathbf{u}_t) - I_h(\mathbf{v}_t) \\ &= I_h(\mathbf{u}_t) + I_h(F(\mathbf{v})) - I_h(\mathbf{f}) - \mu I_h(I_h(\mathbf{w})) + \sum_{k=1}^n \hat{\lambda}_k I_h(G_k(\mathbf{v})), \end{aligned}$$

wherein we have used the linearity of I_h . Defining

$$\boldsymbol{\eta} = I_h(\mathbf{u}_t) + I_h(F(\mathbf{v})) - I_h(\mathbf{f}) - \mu [I_h(I_h(\mathbf{w})) - I_h(\mathbf{w})] + \sum_{k=1}^n \hat{\lambda}_k I_h(G_k(\mathbf{v})),$$

we have

$$(2.2) \quad \frac{d}{dt} I_h(\mathbf{w}) = -\mu I_h(\mathbf{w}) + \boldsymbol{\eta}.$$

This leads to the following convergence result for $I_h(\mathbf{w})$.

PROPOSITION 2.2. *Assume that $\langle I_h(\mathbf{w}), \boldsymbol{\eta} \rangle = 0$ (the inner product is in L^2). Then $I_h(\mathbf{w}) \rightarrow \mathbf{0}$ exponentially in time with decay rate μ .*

Proof. Because $\langle \cdot, \cdot \rangle$ is symmetric and applying (2.2),

$$(2.3) \quad \frac{1}{2} \frac{d}{dt} \|I_h(\mathbf{w})\|^2 = \left\langle \frac{d}{dt} I_h(\mathbf{w}), I_h(\mathbf{w}) \right\rangle = -\mu \|I_h(\mathbf{w})\|^2 + \langle I_h(\mathbf{w}), \boldsymbol{\eta} \rangle,$$

where $\|\mathbf{u}\| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$ is the norm induced by the inner product. By the assumption that $\langle I_h(\mathbf{w}), \boldsymbol{\eta} \rangle = 0$, the previous equation simplifies to

$$\frac{1}{2} \frac{d}{dt} \|I_h(\mathbf{w})\|^2 = -\mu \|I_h(\mathbf{w})\|^2,$$

which has solution $\|I_h(\mathbf{w}(t))\| = \|I_h(\mathbf{w}(0))\| e^{-\mu t}$. \square

Note that $\boldsymbol{\eta}$ depends on time and the parameter estimates $\hat{\lambda}_1, \dots, \hat{\lambda}_n$. Our goal, then, is to choose $\hat{\lambda}_1, \dots, \hat{\lambda}_n$ such that Proposition 2.2 applies and $I_h(\mathbf{w}) \rightarrow \mathbf{0}$. The requisite orthogonality condition is

$$(2.4) \quad \begin{aligned} 0 &= \langle I_h(\mathbf{w}), \boldsymbol{\eta} \rangle \\ &= \langle I_h(\mathbf{w}), I_h(\mathbf{u}_t) + I_h(F(\mathbf{v})) - I_h(\mathbf{f}) \rangle \\ &\quad - \mu \langle I_h(\mathbf{w}), [I_h(I_h(\mathbf{w})) - I_h(\mathbf{w})] \rangle + \sum_{k=1}^n \hat{\lambda}_k \langle I_h(\mathbf{w}), I_h(G_k(\mathbf{v})) \rangle. \end{aligned}$$

As long as at least one of the inner products $\langle I_h(\mathbf{w}), I_h(G_k(\mathbf{v})) \rangle$ is nonzero, we can choose $\hat{\lambda}_1, \dots, \hat{\lambda}_n$ so that (2.4) is satisfied. This is an important assumption: if $I_h(\mathbf{w})$ happens to be orthogonal to $I_h(G_k(\mathbf{v}))$ for every $k = 1, \dots, n$, then the values of $\hat{\lambda}_1, \dots, \hat{\lambda}_n$ have no effect on $\langle I_h(\mathbf{w}), \boldsymbol{\eta} \rangle$.

In the case of a single parameter ($n = 1$), (2.4) uniquely determines the parameter estimate $\hat{\lambda}_1$, provided $\langle I_h(\mathbf{w}), G_1(\mathbf{v}) \rangle \neq 0$. However, if there are multiple parameters ($n > 1$), (2.4) is underdetermined, which prompts us to impose additional constraints. Fix $\mathbf{e}_1 = I_h(\mathbf{w})$ and let $\mathcal{G}(\mathbf{v}) = \text{span}(\{I_h(G_k(\mathbf{v}))\}_{k=1}^n)$. If $\mathcal{G}(\mathbf{v})$ has dimension n (in other words, if the set $\{I_h(G_k(\mathbf{v}))\}_{k=1}^n$ is linearly independent), then we can select $n - 1$ additional linearly independent functions $\mathbf{e}_2, \dots, \mathbf{e}_n \in \mathcal{G}(\mathbf{v})$ that are orthogonal to $\boldsymbol{\eta}$, i.e., $\langle \mathbf{e}_i, \boldsymbol{\eta} \rangle = 0$ for $i = 1, \dots, n$. This requirement gives rise to the $n \times n$ system of equations

$$(2.5) \quad \begin{aligned} A_{i,k} &= \langle \mathbf{e}_i, I_h(G_k(\mathbf{v})) \rangle, \\ \mathbf{A}\hat{\boldsymbol{\lambda}} &= \mathbf{b}, \quad \hat{\boldsymbol{\lambda}} = [\hat{\lambda}_1, \dots, \hat{\lambda}_n]^T, \\ b_i &= \langle \mathbf{e}_i, I_h(\mathbf{f} - \mathbf{u}_t - \mathbf{F}(\mathbf{v})) + \mu [I_h(I_h(\mathbf{w})) - I_h(\mathbf{w})] \rangle, \end{aligned}$$

wherein we have suppressed the dependence on time for simplicity. The assumption that $\dim(\mathcal{G}(\mathbf{v})) = n$ guarantees that this system will have a unique solution.

Remark 2.3. The particular choice of the basis functions $\mathbf{e}_1, \dots, \mathbf{e}_n$ introduces both theoretical and numerical/practical considerations. Proposition 2.2 advocates setting $\mathbf{e}_1 = I_h(\mathbf{w})$ as in (2.4). Theoretically, it may seem advantageous to choose \mathbf{e}_2 so that we can guarantee some higher order norm of $I_h(\mathbf{w})$ decays exponentially, much as selecting $\mathbf{e}_1 = I_h(\mathbf{w})$ leads to decay of $I_h(\mathbf{w})$ in $L^2(\Omega)$. However, it is not clear how to do this in practice and can lead to a numerically expensive calculation, so we instead simply choose the $\mathbf{e}_2, \dots, \mathbf{e}_n$ to form an orthonormal set to obtain a numerically well-conditioned linear system.

Suppose now that \mathbf{w} is small, that is, $\mathbf{v} \approx \mathbf{u}$. In this case, we have

$$\begin{aligned}\boldsymbol{\eta} \approx \tilde{\boldsymbol{\eta}} &:= I_h(\mathbf{u}_t) + I_h(F(\mathbf{u})) - I_h(\mathbf{f}) + \sum_{k=1}^n \hat{\lambda}_k I_h(G_k(\mathbf{u})) \\ &= I_h\left(\mathbf{u}_t + F(\mathbf{u}) - \mathbf{f} + \sum_{k=1}^n \hat{\lambda}_k G_k(\mathbf{u})\right),\end{aligned}$$

provided the operators F and G_1, \dots, G_n are sufficiently smooth.¹ Observe that if the parameter estimates $\hat{\lambda}_1, \dots, \hat{\lambda}_n$ are replaced with the true parameters $\lambda_1, \dots, \lambda_n$, then $\tilde{\boldsymbol{\eta}} \equiv \mathbf{0}$ by (2.1a). This suggests that parameter estimates obtained by solving (2.5) are reasonable approximations to the true parameters when the estimated state \mathbf{v} is close to the true state \mathbf{u} .

Thus, under reasonable assumptions, parameter estimates given by the solution of (2.5) with appropriately selected basis functions $\mathbf{e}_1, \dots, \mathbf{e}_n$ both drive the observable error to zero and at least approximately recover the true parameters when the state estimate converges. These properties are not sufficient to guarantee convergence of the parameter estimates; nevertheless, we will demonstrate that a numerical estimation algorithm based on (2.5) is effective in practice.

2.3. Numerical considerations. As described above, we select $\mathbf{e}_1 = I_h(\mathbf{w})$ and choose the other \mathbf{e}_i to be orthonormal, ensuring that (2.5) is well conditioned. The straightforward approach is then to solve (2.5) at each time step and use the obtained parameter estimates to advance \mathbf{v} . However, this strategy can lead to rapid/discontinuous changes in the parameter at each time step. Such discontinuous adjustments quickly produce numerical instabilities that are unavoidable if the update is applied at each time step, particularly when the initial parameter guess is not close to the true value.

To avoid discontinuous updates of the parameter estimates, we numerically integrate

$$(2.6) \quad \frac{d\hat{\lambda}_k(t)}{dt} = -\alpha(\hat{\lambda}_k(t) - \tilde{\lambda}_k(t)),$$

where $\tilde{\lambda}_k(t)$ is obtained by solving (2.5) at each time t and $\alpha > 0$ is a relaxation parameter. We then solve the coupled system given by (2.1a)–(2.1b) and (2.6). This gives a parameter update that avoids discontinuous changes in the parameters of the system, hence preserving the rigorously justified smoothness of the solutions \mathbf{v} and \mathbf{u} . The effect of the relaxation is examined in Figure 1 for KSE as described in subsection 3.1.1.

¹This is a continuity assumption: if $\mathbf{v} \approx \mathbf{u}$, then $F(\mathbf{v}) \approx F(\mathbf{u})$ and $G_k(\mathbf{v}) \approx G_k(\mathbf{u})$, $k = 1, \dots, n$.

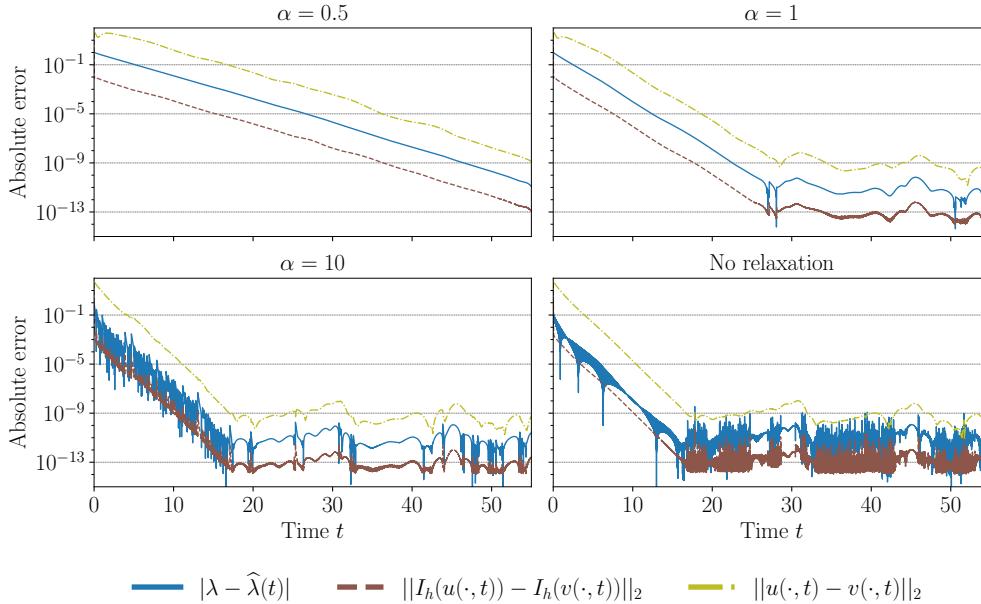


FIG. 1. *Finite relaxation as in (2.6) affects the rate and quality of convergence: strong relaxation (top row) smooths the convergence significantly but slightly decreases the convergence rate (see also Figure 2); mild relaxation (bottom left) preserves the convergence rate that occurs when relaxation is absent (bottom right) but with some smoothing after convergence.*

Algorithm 2.1 details the entire parameter recovery procedure. Note that the algorithm is *concurrent* in the sense that the state estimates $\mathbf{v}_j := \mathbf{v}(t_j)$ can be computed as soon as the projected data at the corresponding time, $\hat{\mathbf{u}}_j := I_h(\mathbf{u}(t_j))$, becomes available. We denote this in the algorithm with the observation operator $\mathcal{Q} : t \mapsto I_h(\mathbf{u}(t))$. The state estimates are updated by time stepping (2.1b), which we denote with a solution operator $\mathcal{S} : (\mathbf{v}_j, \hat{\lambda}_1, \dots, \hat{\lambda}_n) \mapsto \mathbf{v}_{j+1}$ with uniform time step $\delta t = t_{j+1} - t_j$. We make the simplifying assumption that I_h is idempotent, i.e., $I_h(I_h(\mathbf{w})) = I_h(\mathbf{w})$, which slightly simplifies step 15 of the algorithm compared with (2.4)–(2.5). The operators I_h used in our numerical experiments satisfy this property.

Before presenting results of this parameter estimation strategy on KSE, we emphasize that the approach depends critically on assimilating data continuously in time so that the time derivative of the observed state $I_h(\mathbf{u}_t)$ can be approximated accurately. As shown later in Figure 4, the error in the parameter approximation is directly dictated by this finite difference approximation, so that discrete measurements of $I_h(\mathbf{u})$ that are spaced too far apart in time will cause the parameter estimation to break down. All numerical results reported in section 3 are based on continuous data assimilation in the sense that $I_h(\mathbf{u})$ is identified at every time step of the numerical solver.

3. Numerical results. The nondimensionalization of KSE that we consider has the following form:

$$(3.1) \quad u_t + u_{xxxx} + \lambda u_{xx} + uu_x = 0.$$

We impose periodic boundary conditions on the spatial domain $[0, 2\pi L]$. The chaos of the system depends on both L and λ : higher values of L or λ increase the exhibited

Algorithm 2.1 Concurrent multiparameter estimation.

```

1: procedure MPE( Initial time  $t_0 \in \mathbb{R}$ , time step  $\delta t > 0$ , projected state observer
    $\mathcal{Q} : \mathbb{R} \rightarrow \mathcal{V}$ , PDE solver  $\mathcal{S} : \mathcal{V} \times \mathbb{R}^n \rightarrow \mathcal{V}$  (time stepper), initial parameter
   guesses  $\hat{\lambda}_1, \dots, \hat{\lambda}_n \in \mathbb{R}$ , initial state estimate  $\mathbf{v}_0 \in \mathcal{V}$ , finite difference order  $p \in \mathbb{N}$ ,
   projector  $I_h : \mathcal{V} \rightarrow \mathcal{V}$ , nudging parameter  $\mu > 0$ , relaxation parameter  $\alpha > 0$  )
2:    $t \leftarrow t_0$ 
3:   for  $j = 0, 1, \dots$  do
4:     # Observe projected state and estimate its time derivative.
5:      $\hat{\mathbf{u}}_j \leftarrow \mathcal{Q}(t)$  # Projected state observation.
6:      $\hat{\mathbf{u}}'_j \leftarrow \text{BDF}(\hat{\mathbf{u}}_j, \dots, \hat{\mathbf{u}}_{j-p})$  # Backwards difference formula.
7:     # Construct and orthonormalize the basis vectors.
8:      $\mathbf{e}_1 \leftarrow \hat{\mathbf{u}}_j - I_h(\mathbf{v}_j)$ 
9:     for  $i = 2, \dots, n$  do
10:     $\mathbf{e}_i \leftarrow I_h(G_{i-1}(\mathbf{v}_j))$ 
11:     $\mathbf{e}_1, \dots, \mathbf{e}_n \leftarrow \text{MGS}(\mathbf{e}_1, \dots, \mathbf{e}_n)$  # Modified Gram-Schmidt.
12:    # Construct system for estimating parameters.
13:     $\mathbf{b} \in \mathbb{R}^n$ ,  $\mathbf{A} \in \mathbb{R}^{n \times n}$  # Initialize empty  $n \times n$  linear system.
14:    for  $i = 1, \dots, n$  do
15:       $b_i \leftarrow \langle \mathbf{e}_i, I_h(\mathbf{f}) - \hat{\mathbf{u}}'_j - I_h(F(\mathbf{v}_j)) \rangle$ 
16:      for  $k = 1, \dots, n$  do
17:         $A_{i,k} \leftarrow \langle \mathbf{e}_i, I_h(G_k(\mathbf{v}_j)) \rangle$ 
18:    # Get point-in-time estimates and smooth.
19:     $\tilde{\lambda}_1, \dots, \tilde{\lambda}_n \leftarrow \mathbf{A}^{-1}\mathbf{b}$ 
20:    for  $k = 1, \dots, n$  do
21:       $\hat{\lambda}_k \leftarrow (1 + \frac{1}{2}\alpha\delta t)^{-1} \left[ (1 - \frac{1}{2}\alpha\delta t) \tilde{\lambda}_k + \alpha\delta t \tilde{\lambda}_k \right]$ 
22:    # Update estimated state with AOT nudging.
23:     $\mathbf{v}_{j+1} \leftarrow \mathcal{S}(\mathbf{v}_j, \hat{\lambda}_1, \dots, \hat{\lambda}_n) + \mu\delta t(\hat{\mathbf{u}}_j - I_h(\mathbf{v}_j))$ 
24:     $t = t + \delta t$ 
25:   return  $\hat{\lambda}_1, \dots, \hat{\lambda}_n$ 

```

chaos [12, 15, 25, 43]. We choose $L = 16$, $\lambda = 1$ for all simulations as this represents a sufficiently chaotic state [27]. In all simulations, we use a Fourier collocation scheme with a fourth-order Runge–Kutta (RK) implicit-explicit time-stepper [28]. The AOT nudging is applied via a forward Euler time-stepping scheme to alleviate some concerns over the implicit nature of the RK method,² and the parameter smoothing (2.6) is integrated via the trapezoid method. The number of Fourier collocation points is a minimum of 512 in each simulation, which guarantees that even the smallest scales are sufficiently resolved. In most of our experiments, the observation projection operator is a Galerkin truncation onto the lowest 21 modes, meaning that $h \approx \frac{32\pi}{21}$ for all computations reported below. All simulations are run out from $t = 0$ to at least $t_f = 50$ in nondimensional units. The code for running the simulations

²As noted in [37] and observed further in [17], multistage methods such as RK effectively interpolate the data between time steps, introducing some minor interpolation error in the nudging and hence the parameter estimation as well.

and generating the figures in this paper is publicly available at <https://github.com/Parametric-Data-Assimilation/kse-multiparameter-learning>.

3.1. Single-parameter learning. We begin with a discussion of the performance of Algorithm 2.1 when learning a single unknown parameter λ , i.e.,

$$(3.2a) \quad u_t + u_{xxxx} + \lambda u_{xx} + uu_x = 0, \quad (\text{truth})$$

$$(3.2b) \quad v_t + v_{xxxx} + \hat{\lambda} v_{xx} + vv_x = \mu(I_h(u) - I_h(v)), \quad (\text{assim})$$

denoting v for the nudging variable.

3.1.1. Effect of the relaxation and nudging parameters. We generally observe exponential decay of the projected error in time as demonstrated in Figure 1. Defining $w := u - v$, both the solution error $\|w(\cdot, t)\|_2$ and the parameter estimate error $|\lambda - \hat{\lambda}(t)|$ exhibit corresponding exponential decay in time, where $\|\cdot\|_2$ is the L^2 norm over the spatial domain. The decay continues until the error is saturated by the effects of the finite difference time derivative approximation (discussed in subsection 3.1.3), at which point the errors flatten out. Until then, we observe

$$\|w(\cdot, t)\|_2 \approx e^{-\beta t} \|w(\cdot, 0)\|_2,$$

where $\beta > 0$ is the convergence rate. We expect β to depend on the nudging parameter μ of (3.2b) and the relaxation parameter α of (2.6) used in the parameter update. In theory, increasing either μ or α should lead to faster convergence. However, extreme values of either parameter can violate the numerical stability criteria of the selected time-stepping scheme. The natural scaling for μ is with δt^{-1} (the inverse of the time step). The stability region of a simple nudging as applied here indicates that $\mu \geq 2\delta t^{-1}$ leads to instability, so we take $\mu = 1.8\delta t^{-1}$ as the default nudging parameter.

To explore the effect of α on convergence, we fix $\mu = 1.8\delta t^{-1}$ and, for varying values of α , estimate the convergence rate β in two steps: (1) define the time of convergence t_c as the first point in time when the error $\|w(\cdot, t_c)\|_2$ is no better than the error $\|w(\cdot, t_f)\|_2$ at the final time t_f , and then (2) estimate β from the time of convergence t_c and the logarithmic ratio of the initial error and the error at t_c , i.e.,

$$\beta \approx -\frac{\log \|w(\cdot, t_c)\|_2 - \log \|w(\cdot, t_0)\|_2}{t_c - t_0}.$$

Figure 2 shows that β generally increases with α until about $\alpha = 10$. Repeating the experiment with fixed $\alpha = 1$ and varying μ shows that scaling μ down to $\mu < \mathcal{O}(10)$ decreases the convergence rate, but the dependence of β on μ is not as significant as the dependence on α . In addition to affecting the convergence rate, α has a strong effect on the smoothness of the convergence, though this is more difficult to precisely quantify. Figure 1 shows the convergence for a few choices of α , with $\mu = 1.8\delta t^{-1}$. For single-parameter estimation, spikes in the convergence without relaxation as seen in the bottom right plot of Figure 1 are not concerning, but such discontinuous jumps in the parameters can be problematic when estimating multiple parameters. Based on these observations, we set $\alpha = 1$ and $\mu = 1.8\delta t^{-1}$ for all subsequent experiments as a compromise between the speed and the smoothness of convergence.

3.1.2. Choice of observation operator. Up to this point, for the observation operator I_h our experiments use a Fourier projection that truncates the state to the lowest 21 Fourier modes. This particular choice is natural given that we take a spectral approach to numerically solving (3.2a)–(3.2b) (see Appendix A). Figure 3

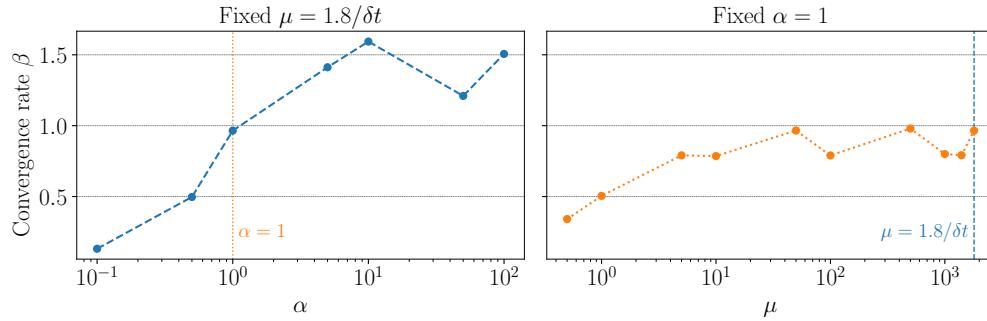


FIG. 2. The convergence rate depends on the smoothing parameter α (left) and the assimilation constant μ (right). However, faster convergence is accompanied by nonsmooth decay of the error in the parameter estimate (see Figure 1).

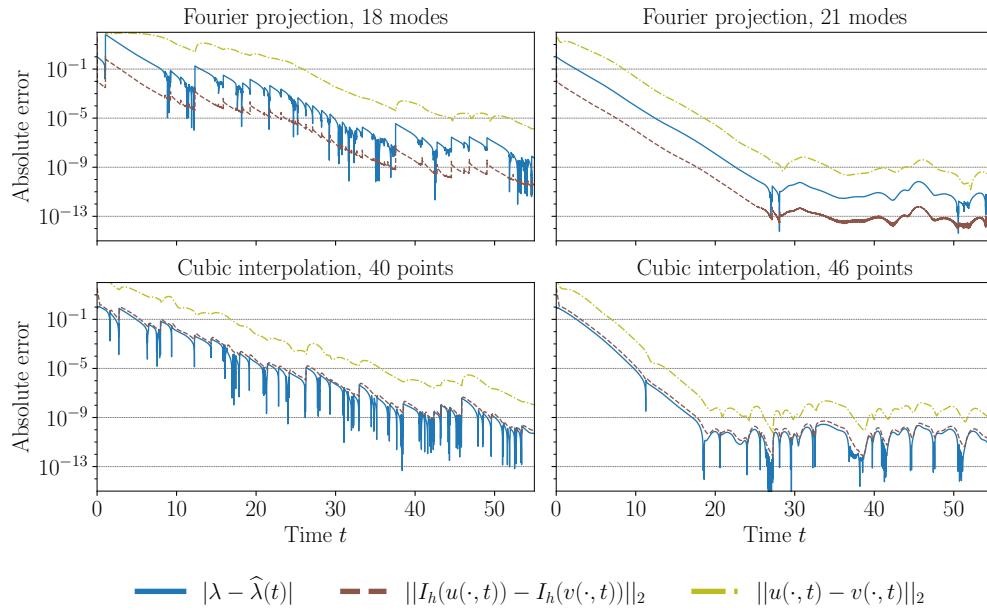


FIG. 3. Convergence of single-parameter learning with Fourier truncation for the projection operator I_h (top) and a cubic spline interpolator for I_h (bottom). The plots on the left represent the minimal number of observations required for synchronization in each strategy.

shows results comparable to Figure 1 but with only 18 observed Fourier modes, which exhibits convergence but with less smoothness and at a slightly slower rate. Using fewer than 18 Fourier modes does not result in synchronization.

To further validate our method for a wider class of physical settings, we experiment with observation operators I_h based on pointwise observations at evenly spaced points in the domain. Specifically, Figure 3 shows results when using cubic spline interpolation with 40 and 46 observed points. While not shown, using a linear or quadratic interpolation for I_h yields similar results. The default nudging parameter $\mu=1.8\delta t^{-1}$ does not result in synchronization in these experiments, and consequently we use the slightly more conservative value $\mu = 10$ in conjunction with pointwise observations. Even so, the convergence in the pointwise observations case is similar to

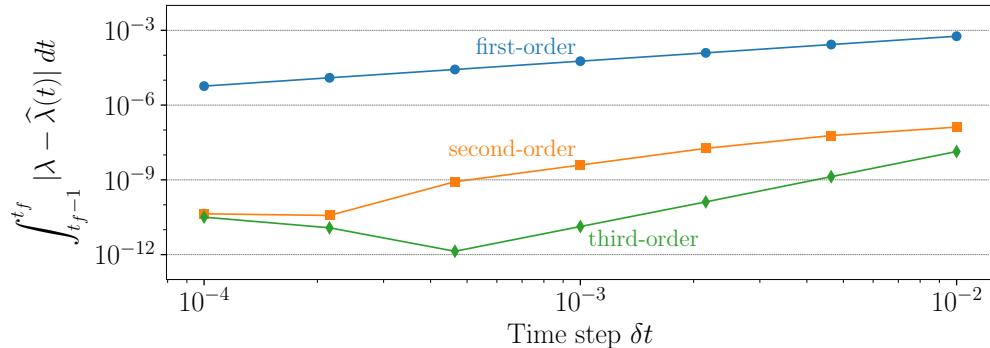


FIG. 4. Error in the parameter estimate with respect to the time step δt using first-order, second-order, and third-order backward finite difference schemes to estimate the projected time derivative $I_h(u_t)$. The order of the error is 1.0004, 1.9489, and 3.0023, respectively (for the third-order method we exclude the two leftmost points). Excluding the leftmost point for the second-order finite differences raises the order estimate to 2.0535.

the Fourier projection case, although the number of pointwise observations required to obtain convergence (40) is just over twice the number of Fourier modes needed (18). At these marginal observation levels the convergence is rough, although adding just a few more observations results in a much smoother synchronization. On the other hand, significantly increasing the number of observations beyond this marginal threshold does not substantially improve either the final total error or the convergence rate.

3.1.3. Order of accuracy. The accuracy of the parameter estimate, and hence the estimate of the solution u , is limited by three factors: (1) the error introduced by the numerical scheme for the PDE, (2) round-off error, and (3) errors introduced by finite difference estimates of $I_h(u_t)$. Of these factors, the finite difference error for estimating $I_h(u_t)$ is the most significant, and therefore the order of the corresponding finite difference scheme controls the overall accuracy of the final parameter estimate with respect to the time step δt . Figure 4 shows the average error in the parameter estimate over the final simulation second as a function of δt , with results using first-order, second-order, and third-order backward difference methods to estimate $I_h(u_t)$ (we use only backward differences as only data at the current and previous time steps are available). We note that errors from the underlying PDE solver, and particularly the data assimilation aspect of the time stepping with the RK scheme,³ limit the overall error to approximately 10^{-12} , which is seen by the saturation of the error for both the second- and third-order finite difference methods in Figure 4. Omitting these final points, we obtain a linear regressed exponent that fits nearly exactly with the order of the finite difference method for the temporal derivative. As these numerical investigations are meant to provide a proof of concept only, the error introduced via the RK time stepping is an acceptable compromise to facilitate ease of implementation and accessibility of the code.

We consider the propagation of the error in the third-order backward difference method specifically to understand the increase in the parameter error for $\delta t < 10^{-3}$. If we allow ξ to be the error from the RK time-stepping in the data assimilation step,

³See [17] for a description of this kind of error in a different setting and [37] for a more thorough discussion of this issue.

then the numerically “converged” state of the system should be $\tilde{I}_h(u) \approx I_h(u) + \xi$, where $\xi = \xi(\delta t, N, \epsilon_{\text{machine}})$ is a function of the time step size, number of collocation points N , and machine epsilon $\epsilon_{\text{machine}}$. Using a third-order backward difference to approximate $I_h(u_t)$ in this case leads to

$$\tilde{I}_h(u_t) \approx I_h(u_t) + \frac{\xi}{\delta t} + \mathcal{O}(\delta t^3).$$

The issue is that, as noted in [17] in a different setting, the multistage RK method introduces an error $\xi > \epsilon_{\text{machine}}$ independent of δt itself. Thus, there is a certain point at which decreasing δt counterintuitively leads to increased error in $\tilde{I}_h(u_t)$ and hence an increased error in the parameter estimation. The rub of the matter is that the two smallest values of δt for the third-order difference in Figure 4 are purely artifacts of the RK time stepping and are hence numerically irrelevant to a discussion on the order of convergence.

3.2. Multiparameter learning. The primary contribution of Algorithm 2.1 is the ability to learn multiple unknown parameters in a single equation in a concurrent fashion. To this end, we consider a *generalized KSE* for the assimilating state $v(x, t)$, defined as

$$(3.3a) \quad u_t + \lambda_4 u_{xxxx} + \lambda_2 u_{xx} + \lambda_5 uu_x = 0, \quad (\text{truth})$$

$$(3.3b) \quad v_t + \hat{\lambda}_4 v_{xxxx} + \hat{\lambda}_3 v_{xxx} + \hat{\lambda}_2 v_{xx} + \hat{\lambda}_1 v_x + \hat{\lambda}_5 vv_x = \mu(I_h(u) - I_h(v)). \quad (\text{assim})$$

Once again we consider $x \in [0, 2\pi L]$ with $L = 16$ fixed and run all simulations from time $t = 0$ to $t_f = 50$. Following the discussion above for the single-parameter case, we set $\delta t = 10^{-3}$, $\mu = 1.8\delta t^{-1}$, $\alpha = 1$, define the projection operator I_h as the Fourier projection that truncates to the lowest 21 modes, and use third-order backward differences to approximate the projection of the temporal derivative $I_h(u_t)$.

The goal of this investigation is not only to identify the correct parameters of the specified PDE but also to discover the structure of an unknown equation. Hence, we have expanded the standard KSE to include unknown parameters on all terms and additional parameters on linear odd-order derivatives. The true KSE corresponds to the case when $\lambda_1 = \lambda_3 = 0$ and $\lambda_2 = \lambda_4 = \lambda_5 = 1$. In each experiment, we choose a subset of the λ_k to learn and set the rest to their true values. The simulation is initialized with an initial guess of $\hat{\lambda}_k = 2$ for every λ_k that is unknown.

We focus here on identifying the coefficient λ_5 in front of the known quadratic nonlinearity uu_x , although the derivation of Algorithm 2.1 presented above technically holds for additional nonlinear terms. The reason for this restriction in the current setting is that adding other nonlinear terms to KSE destabilizes the dynamical evolution so that neither the original system nor the nudged system can be simulated easily. Hence, the restriction to a single nonlinear term here is more a product of the specific dynamical system we are investigating (KSE) than a product of the proposed algorithm.

We preface our numerical results by first discussing a few implementation details which are not obvious from the theoretical derivation of the algorithm.

3.2.1. Choice of basis. A crucial piece of the algorithm is the choice of basis functions e_1, \dots, e_N . Other than setting $e_1 = I_h(w)$ (to ensure exponential decay of $I_h(w)$ in L^2), it is not clear how to choose these functions. For numerical efficiency, we would like to use information that has already been computed. A natural choice is then to use the $I_h(G_k(v))$, as these are computed in the simulation itself. Here

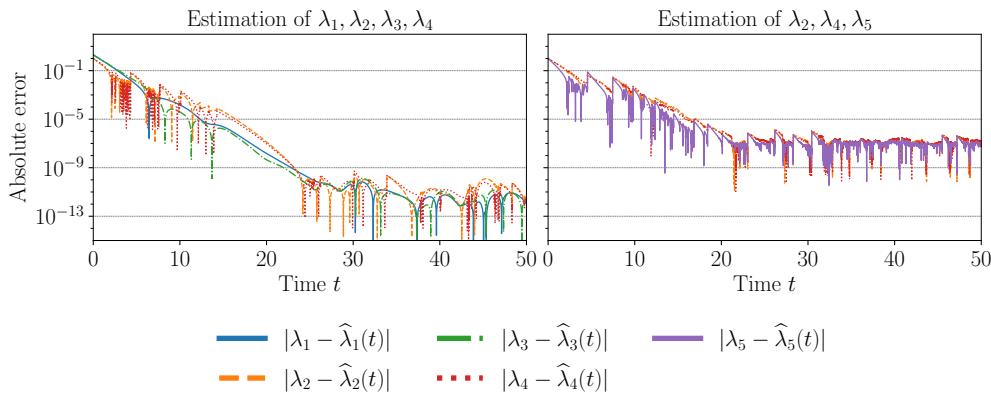


FIG. 5. Convergence of multiparameter estimation with respect to time for the four parameters corresponding to linear terms in the generalized KSE (3.3b) (left) and for the three parameters corresponding to terms in the original KSE, including the coefficient λ_5 on the nonlinear term uu_x (right).

another ambiguity arises—we need only $n - 1$ more basis elements, but there are n of the $I_h(G_k(v))$; hence we may exclude one. Most of the time, the choice does not affect convergence. However, we find that when G_k corresponds to the nonlinear term in the generalized KSE, including it as a basis element causes the algorithm to not converge. Consequently, we always exclude this term from the basis.

For numerical stability, the basis vectors are orthonormalized in L^2 before constructing the system (2.5). While it is technically possible for the system to be degenerate, we do not observe this in practice at least in this setting.

3.2.2. Convergence. As in the case of single-parameter estimation, we observe exponential convergence in time to the true parameters in the multiparameter case. However, there are additional limitations when multiple parameters are considered. When estimating multiple unknown parameters, the point-in-time estimates must be smoothed as in (2.6) or the algorithm will fail to converge; i.e., the rapid changes in parameter values introduce nonphysical discontinuous features and reactions to the system which overwhelm the continuous data assimilation. Additionally, not every combination of unknown parameters can be learned (at least based on the experiments considered here). In particular, we find that λ_5 (the coefficient for the nonlinear term) cannot be learned at the same time as λ_1 or λ_3 —both of which are equal to 0 in the true KSE. Every other combination of unknown parameters can be successfully recovered. In other words, it appears that this approach can either recover the nonlinear dynamics, or it can perform model discovery, but not both simultaneously.

Estimation of the coefficients of the linear terms $\lambda_1, \lambda_2, \lambda_3$, and λ_4 is very efficient and appears to work down to an error of $\mathcal{O}(10^{-11})$, consistent with our previous results when recovering a single parameter. See Figure 5 for convergence of all four linear-term coefficients with $\hat{\lambda}_5(t) \equiv \lambda_5 = 1$ known. Note that the convergence rate is very similar to that observed for the single parameter case (compare with the top right plot of Figure 1). It also appears that estimation of the dissipative coefficients ($k = 2, 4$) behave similarly to one another, while the dispersive terms ($k = 1, 3$) converge in a similar fashion (smoother than the dissipative ones). This could either be a consequence of the differences between dispersion and dissipation, or it may be simply because $\lambda_1 = \lambda_3 = 0$ in the true system.

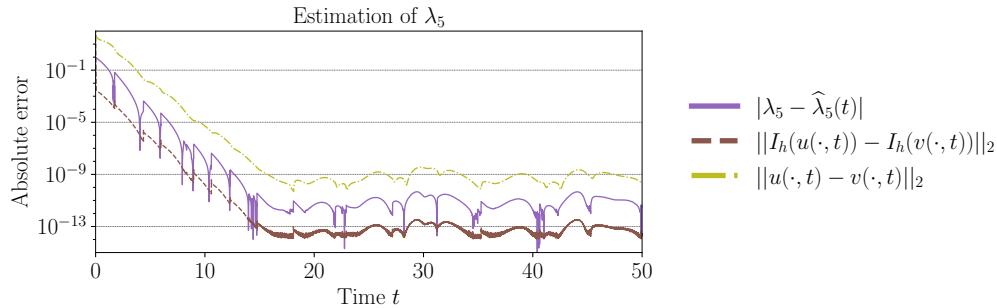


FIG. 6. Convergence of single-parameter estimation with respect to time for the parameter λ_5 corresponding to the nonlinear term uu_x only in KSE, as well as the state error and the projected state error.

Complete parameter recovery without the addition of the dispersive $k = 1, 3$ terms, but including the nonlinear term, is also demonstrated in Figure 5. Note that the final “converged” error in this case is significantly worse, appearing to level off at $\mathcal{O}(10^{-7})$ for all parameters, even though the convergence rate is very similar (the error stops decreasing at about the same time). It is not immediately clear why this combination of coefficients does not achieve the same level of accuracy, but it appears that the unknown coefficient on the nonlinearity plays a pivotal role. Even with this decrease in final accuracy, recovery of the true parameters up to $\mathcal{O}(10^{-7})$ for such a concurrent learning algorithm is significant.

It is worth mentioning that the algorithm experiences no loss of accuracy when λ_5 is the single unknown parameter, as reported in Figure 6. A decrease in accuracy only arises when estimating λ_5 in tandem with other unknown parameters. This indicates that there is a subtle interplay between recovery of the nonlinear coefficient and recovery of coefficients for the linear terms.

Our experiments indicate that the method appears to be robust to changes in the values of α and/or the initial values of the parameters; i.e., we may make a choice other than $\hat{\lambda}_k(0) = 2$ with little effect on the convergence of $\hat{\lambda}_k(t)$ to λ_k . On the contrary, the choice of the basis functions $\{e_i\}_{i=1}^N$ is critical: failing to include the error term $e_1 = I_h(w)$ and instead using basis functions corresponding to the first n Fourier modes leads to a failure to converge. We also implemented an alternative approach that estimated the parameters at each time-step via a least-squares fit using several additional basis functions (N total), as opposed to exactly solving a small linear system. This led to similar convergence to the method presented in detail here but at slightly increased computational expense. Furthermore, the theoretical justification is less clear for this alternate method, so we leave a more detailed investigation of this and other alternative basis functions to a later study.

4. Conclusions. We have developed an algorithm for concurrently learning scalar parameters and assimilating the state in a general setting for evolutionary, dissipative PDEs and demonstrated that the method works remarkably well on a generalized version of the KSE. Numerically, we observe that multiple parameters (coefficients) of KSE can be recovered up to an explainable error (due to the RK time stepping and data assimilation method) so long as these parameters are coefficients on linear terms in the equation. We have also demonstrated that the coefficient of the nonlinear term can be recovered but only in concert with dissipative linear terms and then at a nontrivial increase to the final error. Although there are limitations to the

developed approach, including the need for a high temporal resolution, it is shown to work remarkably well on KSE.

Further applications of this algorithm are readily apparent. Parameter learning or model recovery for physically motivated systems are interesting and well-motivated problems. Although the derivation in section 2 is not yet rigorous, it does appear that this approach will work for all dissipative systems for which the AOT algorithm has been shown to converge, which represents a very large class of PDEs. Extensions beyond the simplified setting where the “true” system is run simultaneously as a direct numerical simulation are of greater interest and will be explored at length. For instance, a lofty extension of this work would be to estimate the skin friction of a surface from data collected via wind tunnel experiments. Such an application of this approach of course entails additional complications that have not been considered here.

Finally, though the derivation provided here is intuitive, it is not yet fully rigorous. Specifically, we have not provided rigorous justification for the observed convergence of both the parameter learning and the simultaneous synchronization of the state. The rigorous justification of this algorithm for certain systems of PDEs will not only provide a firm mathematical foundation for this effort but will also highlight the necessary hypotheses on the hyperparameters (quality of the observations defined by the parameter h for example) that will guide the practical implementation of this method in more practical settings.

Appendix A. Numerical solution to the generalized KSE. This section provides essential details on the KSE numerical solver.

We consider a generalized version of KSE,

$$u_t + \lambda_1 u_x + \lambda_2 u_{xx} + \lambda_3 u_{xxx} + \lambda_4 u_{xxxx} + \lambda_5 u u_x = 0.$$

The classical KSE has $\lambda_1 = \lambda_3 = 0$. Let \mathcal{F} denote the Fourier transform and denote $\hat{u}(\xi, t) = [\mathcal{F}(u(\cdot, t))](\xi)$. Recall $[\mathcal{F}(\frac{d^k}{dx^k} \phi)](\xi) = (2\pi i \xi)^k [\mathcal{F}(\phi)](\xi)$. Taking the Fourier transform of both sides of the generalized KSE we arrive at the spectral version of the PDE,

$$\hat{u}_t + \lambda_1(2\pi i \xi) \hat{u} + \lambda_2(2\pi i \xi)^2 \hat{u} + \lambda_3(2\pi i \xi)^3 \hat{u} + \lambda_4(2\pi i \xi)^4 \hat{u} + \lambda_5 \mathcal{F}(u u_x) = 0.$$

To elucidate the numerical implementation of this equation, we let \mathcal{F}^{-1} denote the inverse Fourier transform, i.e., $u(x, t) = \mathcal{F}^{-1}(\hat{u})$. This immediately yields the standard pseudospectral scheme:

$$\begin{aligned} \hat{u}_t = & - [\lambda_1(2\pi i \xi) + \lambda_2(2\pi i \xi)^2 + \lambda_3(2\pi i \xi)^3 + \lambda_4(2\pi i \xi)^4] \hat{u} \\ & - \lambda_5 \mathcal{F}(\mathcal{F}^{-1}(\hat{u}) \mathcal{F}^{-1}(2\pi i \xi \hat{u})). \end{aligned}$$

The primary remaining aspects of the numerical scheme are then the choice of a time-stepping discretization of \hat{u}_t and the finite truncation level of the Fourier transform.

All the simulations performed here are done with an implicit-explicit RK four-stage method that is formally fourth-order, and the Fourier series are all truncated with $N = 512$ spatial points. We also implement a 2/3 dealiasing to avoid unnecessary overflow to the smallest scales. The choice of domain size and the value of $\lambda_1 = 1$ is selected for this degree of spatial resolution because even when dealiasing is removed, the power spectrum of the simulation is fully resolved with these parameters, i.e., the magnitude of the smallest-scale terms is near $\epsilon_{\text{machine}}$.

For the data assimilation step, the projection operator $I_h(u)$ is defined by zeroing out all modes with wave numbers greater than a cutoff K selected as $K = 21$ for all results throughout the paper (except as noted in subsection 3.1.2 and Figure 3). All simulations for the true state are started with initial condition

$$u_0(x) = \sin\left(6\frac{\pi x}{L}\right) + 0.1 \cos\left(\frac{\pi x}{L}\right) - 0.2 \sin\left(3\frac{\pi x}{L}\right) \\ + 0.05 \cos\left(15\frac{\pi x}{L}\right) + 0.7 \sin\left(18\frac{\pi x}{L}\right) - \cos\left(13\frac{\pi x}{L}\right)$$

and then run out to nondimensional time $\tilde{t} = 10$ whereupon the assimilated system is initialized.

Acknowledgments. The authors would like to thank E. Carlson, J. Hudson, A. Larios, V. Martinez, and E. Ng for their helpful discussions, as well as the reviewers for their very helpful feedback which led to significant improvements in the manuscript.

REFERENCES

- [1] A. AZOUANI, E. OLSON, AND E. S. TITI, *Continuous data assimilation using general interpolant observables*, J. Nonlinear Sci., 24 (2014), pp. 277–304.
- [2] H. BESSAIH, E. OLSON, AND E. S. TITI, *Continuous data assimilation with stochastically noisy data*, Nonlinearity, 28 (2015), p. 729.
- [3] A. BISWAS, Z. BRADSHAW, AND M. S. JOLLY, *Data assimilation for the Navier-Stokes equations using local observables*, SIAM J. Appl. Dyn. Syst., 20 (2021), pp. 2174–2203.
- [4] A. BISWAS, J. HUDSON, A. LARIOS, AND Y. PEI, *Continuous data assimilation for the 2D magnetohydrodynamic equations using one component of the velocity and magnetic fields*, Asymptot. Anal., 108 (2018), pp. 1–43.
- [5] J. BORGGAARD, N. GLATT-HOLTZ, AND J. KROMETIS, *A Bayesian approach to estimating background flows from a passive scalar*, SIAM/ASA J. Uncertain. Quantif., 8 (2020), pp. 1036–1060.
- [6] S. L. BRUNTON, J. L. PROCTOR, AND J. N. KUTZ, *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*, Proc. Natl. Acad. Sci. USA, 113 (2016), pp. 3932–3937.
- [7] N. B. BUDANUR AND P. CVITANOVIC, *Unstable manifolds of relative periodic orbits in the symmetry-reduced state space of the Kuramoto-Sivashinsky system*, J. Stat. Phys., 167 (2017), pp. 636–655.
- [8] E. CARLSON, J. HUDSON, AND A. LARIOS, *Parameter recovery for the 2 dimensional Navier-Stokes equations via continuous data assimilation*, SIAM J. Sci. Comput., 42 (2020), pp. A250–A270.
- [9] E. CARLSON, J. HUDSON, A. LARIOS, V. R. MARTINEZ, E. NG, AND J. P. WHITEHEAD, *Dynamically learning the parameters of a chaotic system using partial observations*, Discrete Continuous Dyn. Syst., 42 (2022), pp. 3809–3839.
- [10] B. I. COHEN, J. KROMMES, W. TANG, AND M. ROSENBLUTH, *Non-linear saturation of the dissipative trapped-ion mode by mode coupling*, Nuclear Fusion, 16 (1976), p. 971.
- [11] P. COLLET, J.-P. ECKMANN, H. EPSTEIN, AND J. STUBBE, *A global attracting set for the Kuramoto-Sivashinsky equation*, Comm. Math. Phys., 152 (1993), pp. 203–214.
- [12] P. CVITANOVIC, R. L. DAVIDCHACK, AND E. SIMINOS, *On the state space geometry of the Kuramoto-Sivashinsky flow in a periodic domain*, SIAM J. Appl. Dyn. Syst., 9 (2010), pp. 1–33.
- [13] M. DASHTI AND A. M. STUART, *The Bayesian approach to inverse problems*, in Handbook of Uncertainty Quantification, R. Ghanem, D. Higdon, H. Owhadi, eds, Springer, Cham, 2016, pp. 311–428.
- [14] P. C. DI LEONI, A. MAZZINO, AND L. BIFERALE, *Inferring flow parameters and turbulent configuration with physics-informed data assimilation and spectral nudging*, Phys. Rev. Fluids, 3 (2018), 104604.
- [15] R. A. EDSON, J. E. BUNDER, T. W. MATTNER, AND A. J. ROBERTS, *Lyapunov exponents of the Kuramoto-Sivashinsky PDE*, ANZIAM J., 61 (2019), pp. 270–285.
- [16] L. C. EVANS, *Partial Differential Equations*, Grad. Stud. Math. 19, American Math Society, Providence, RI, 1998.

- [17] A. FARHAT, N. GLATT-HOLTZ, V. R. MARTINEZ, S. A. MCQUARRIE, AND J. P. WHITEHEAD, *Data assimilation in large Prandtl Rayleigh–Bénard convection from thermal measurements*, SIAM J. Appl. Dyn. Syst., 19 (2020), pp. 510–540.
- [18] A. FARHAT, H. JOHNSTON, M. JOLLY, AND E. S. TITI, *Assimilation of nearly turbulent Rayleigh–Bénard flow through vorticity or local circulation measurements: A computational study*, J. Sci. Comput., (2017), pp. 1–15.
- [19] A. FARHAT, M. S. JOLLY, AND E. S. TITI, *Continuous data assimilation for the 2D Bénard convection through velocity measurements alone*, Phys. D Nonlinear Phenom., 303 (2015), pp. 59–66.
- [20] A. FARHAT, E. LUNASIN, AND E. S. TITI, *Abridged continuous data assimilation for the 2D Navier–Stokes equations utilizing measurements of only one component of the velocity field*, J. Math. Fluid Mech., 18 (2016), pp. 1–23.
- [21] A. FARHAT, E. LUNASIN, AND E. S. TITI, *On the Charney conjecture of data assimilation employing temperature measurements alone: The paradigm of 3D planetary geostrophic model*, Mathematics of Climate and Weather Forecasting, 2 (2016).
- [22] C. FOIAS, G. R. SELL, AND E. S. TITI, *Exponential tracking and approximation of inertial manifolds for dissipative nonlinear equations*, J. Dyn. Differential Equations, 1 (1989), pp. 199–244.
- [23] D. GOLUSKIN AND G. FANTUZZI, *Bounds on mean energy in the Kuramoto–Sivashinsky equation computed using semidefinite programming*, Nonlinearity, 32 (2019), p. 1705.
- [24] Z. GRUJIĆ, *Spatial analyticity on the global attractor for the Kuramoto–Sivashinsky equation*, J. Dyn. Differential Equations, 12 (2000), pp. 217–228.
- [25] J. M. HYMAN AND B. NICOLAENKO, *The Kuramoto–Sivashinsky equation: A bridge between PDE’s and dynamical systems*, Phys. D Nonlinear Phenom., 18 (1986), pp. 113–126.
- [26] M. S. JOLLY, V. R. MARTINEZ, AND E. S. TITI, *A data assimilation algorithm for the subcritical surface quasi-geostrophic equation*, Adv. Nonlinear Stud., 17 (2017), pp. 167–192.
- [27] A.-K. KASSAM AND L. N. TREFETHEN, *Fourth-order time-stepping for stiff PDEs*, SIAM J. Sci. Comput., 26 (2005), pp. 1214–1233.
- [28] C. A. KENNEDY AND M. H. CARPENTER, *Additive Runge–Kutta schemes for convection–diffusion–reaction equations*, Appl. Numer. Math., 44 (2003), pp. 139–181.
- [29] Y. KURAMOTO AND T. TSUZUKI, *On the formation of dissipative structures in reaction-diffusion systems: Reductive perturbation approach*, Prog. Theor. Phys., 54 (1975), pp. 687–699.
- [30] R. E. LAQUEY, S. MAHAJAN, P. RUTHERFORD, AND W. TANG, *Nonlinear saturation of the trapped-ion mode*, Phys. Rev. Lett., 34 (1975), p. 391.
- [31] A. LARIOS AND Y. PEI, *Nonlinear Continuous Data Assimilation*, preprint, arXiv:1703.03546, 2017.
- [32] Z. LI, N. KOVACKI, K. AZIZZADENESHeli, B. LIU, K. BHATTACHARYA, A. STUART, AND A. ANANDKUMAR, *Fourier Neural Operator for Parametric Partial Differential Equations*, preprint, arXiv:2010.08895, 2020.
- [33] E. N. LORENZ, *Deterministic nonperiodic flow*, J. Atmos. Sci., 20 (1963), pp. 130–141.
- [34] L. LU, P. JIN, G. PANG, Z. ZHANG, AND G. E. KARNIADAKIS, *Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators*, Nat. Mach. Intell., 3 (2021), pp. 218–229.
- [35] C. MISBAH AND A. VALANCE, *Secondary instabilities in the stabilized Kuramoto–Sivashinsky equation*, Phys. Rev. E, 49 (1994), p. 166.
- [36] B. NICOLAENKO, B. SCHEURER, AND R. TEMAM, *Some global dynamical properties of the Kuramoto–Sivashinsky equations: Nonlinear stability and attractors*, Phys. D Nonlinear Phenom., 16 (1985), pp. 155–183.
- [37] E. OLSON AND E. S. TITI, *Determining modes and Grashof number in 2D turbulence: A numerical case study*, Theor. Comput. Fluid Dyn., 22 (2008), pp. 327–339.
- [38] M. QUADE, M. ABEL, J. NATHAN KUTZ, AND S. L. BRUNTON, *Sparse identification of nonlinear dynamics for rapid model recovery*, Chaos, 28 (2018), 063116.
- [39] M. RAISSI, P. PERDIKARIS, AND G. KARNIADAKIS, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, J. Comput. Phys., 378 (2019), pp. 686–707.
- [40] S. RUDY, A. ALLA, S. L. BRUNTON, AND J. N. KUTZ, *Data-driven identification of parametric partial differential equations*, SIAM J. Appl. Dyn. Syst., 18 (2019), pp. 643–660.
- [41] G. SIVASHINSKY, *Nonlinear analysis of hydrodynamic instability in laminar flames– I. Derivation of basic equations*, Acta Astronaut., 4 (1977), pp. 1177–1206.
- [42] G. I. SIVASHINSKY AND D. MICHELSON, *On irregular wavy flow of a liquid film down a vertical plane*, Prog. Theor. Phys., 63 (1980), pp. 2112–2114.
- [43] R. W. WITTENBERG AND P. HOLMES, *Scale and space localization in the Kuramoto–Sivashinsky equation*, Chaos, 9 (1999), pp. 452–465.